SPATIO-TEMPORAL OPTIMIZATION FOR CONTROL OF INFINITE DIMENSIONAL SYSTEMS IN ROBOTICS, FLUID MECHANICS, AND QUANTUM MECHANICS

A Dissertation
Presented to
The Academic Faculty

By

Ethan N. Evans

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
Daniel Guggenheim School of Aerospace Engineering
Department of Aerospace Engineering

Georgia Institute of Technology

August 2021

SPATIO-TEMPORAL OPTIMIZATION FOR CONTROL OF INFINITE DIMENSIONAL SYSTEMS IN ROBOTICS, FLUID MECHANICS, AND QUANTUM MECHANICS

Thesis committee:

Dr. Evangelos Theodorou
Department of Aerospace Engineering
Georgia Institute of Technology

Dr. Andrzej Swiech
Department of Mathematics
Georgia Institute of Technology

Dr. Kyriakos Vamvoudakis
Department of Aerospace Engineering
Georgia Institute of Technology

Dr. Matthew Bays
Senior Research Scientist
Naval Surface Warfare Center – Panama City Division

Dr. Yongxin Chen
Department of Aerospace Engineering
Georgia Institute of Technology

Dr. Michael DeWeese
Department of Physics
University of California, Berkeley

Date approved: July 26, 2021

We are at the very beginning of time for the human race. It is not unreasonable that we grapple with problems. But there are tens of thousands of years in the future.
Our responsibility is to do what we can, learn what we can,
improve the solutions, and pass them on.

Richard P. Feynman

Dedicated to my mother, who did not complete her doctorate dream and instead devoted every waking minute to having and raising me and my siblings. May the successes of your children be your own successes, and may this thesis in some way serve as a fulfillment of your long-lost dream.

ACKNOWLEDGMENTS

First and foremost I would like to thank my PhD Advisor, Evangelos Theodorou. You have always pushed me beyond what I thought I could achieve, exposed me to deep ideas I had not considered, and helped me inspire confidence within myself. Thank you for being my constant friend and mentor, helping me through hardships, guiding my academic ideas, and finally enabling me to reach this moment. From day one you saw merit in my fantasies about working with high risk quantum systems, reminded me of my passion even when we ran into walls, and encouraged me to constantly revisit the quantum applications that now motivate my career. You constantly teach me, and all of your students, that there truly are no boundaries in science, you inspire us to have atypical perspectives, and you give us the confidence to never stop pursuing our dreams.

I would like to express my deep gratitude to my thesis committee for their help throughout this process. Andrzej Swiech, our conversations together have been extremely enlightening, and helped us resolve several issues with the formulation. Mike DeWeese, working with you and your student Adam Frim has been an absolute pleasure, and has enabled me to truly appreciate the physics of the quantum control problems in a profound way. Yongxin Chen, your pragmatism and detailed questions helped me refine my notation and see my work from a different perspective. Kyriakos Vamvoudakis, your positivity, curiosity, and novel perspective helped me tremendously, especially regarding proofs of convergence. Last but certainly not least, Matt Bays. Thank you for guiding me and mentoring me for over five years. Thank you for always believing in me, for hiring me for an internship, for helping me get SMART funding, and for constantly pushing me to take my ideas to impactful real world applications. Thank you for constantly looking out for me and for always finding new ways for me to propel my ideas and my upcoming DoD career forward.

I would also like to thank my other mentors and sources of funding throughout the PhD process. Jechiel Jagoda, thank you for always listening to my kvetches, giving me guidance

when I was lost, for finding me GTAs when I didn't have GRA funding, and for always looking out for me. It was always somehow magical that after coming to your office, my issues always seemed to work out soon after. Dimitri Mavris, thank you for providing me with GRA funding for two years, introducing me to the world of naval robotics, and treating me as one of your own in the ASDL environment. I will always appreciate the scientific methodology that you take with your students, and hope that this is somehow reflected in my own work and philosophy. Finally, I would also like to gratefully acknowledge the support from the Science, Mathematics, and Research for Transformation (SMART) scholarship for the last 3 years of my PhD. This scholarship gave me the flexibility and independence needed to pursue high risk ideas and applications.

I would also like to thank my family. My parents Phil and Vivian have always put their children first, and have provided every advantage they could imagine for me and my siblings. They managed to kindle my spark of curiousity from a very young age, and have supported me unconditionally at every life stage. My older brother, Ari, has always been my shining example of work ethic and enduring persistence. My little sister, Talia, teaches all around her to make sound arguments, and has unmatched creativity. Hopefully signs of these traits may be found throughout the pages of this thesis, and in every case are thanks to their constant positive impact in my life. I would also like to thank my friends, co-authors, and colleagues that helped me develop myself and the ideas in this thesis, including Patrick Meyer, Coline Ramee, Sam Seifert, Marcus Pereira, George Boutselis, Andrew Kendall, Adam Frim, Edan Baltman, Jason Mizrahi, Adam Gerson, Max Gorman, and Shlomoh and Shifra Sharfstein.

Finally, and perhaps most of all, I would like to thank my loving wife, Shani. You give me constant support, dedication, and encouragement. You kept me calm and collected through the PhD qualifying exam 'experience', you find every way to fill a stressful time with laughter, you constantly encourage me after every bump in the road, and you help me in every way possible so that I can focus on my deadlines. You've helped me in unquantifiable ways to reach this summit, and you have truly made this all possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

1D 1-dimensional

2D 2-dimensional

3D 3-dimensional

ADPL Actuator Design and Policy Learning

ANN Artificial Neural Network

CFD Computational Fluid Dynamics

CNN Convolutional Neural Network

DDP Differential Dynamic Programming

DL Deep Learning

FC Fully Connected

FLOP Floating Point Operation

FNN Feed-forward Neural Network

GASS Gradient-based Adaptive Stochastic Search

GRAPE Gradient Ascent Pulse Engineering

HJB Hamilton-Jacobi-Bellman

IDVRL Infinite Dimensional Variational Reinforcement Learning

ITC Information Theoretic Control

KL Kullback-Leibler

LQR Linear Quadratic Regulator

MPC Model Predictive Control

MPPI Model Predictive Path Integral

NODE Neural Ordinary Differential Equation

NP Non-Polynomial-Time

ODE Ordinary Differential Equation

PDE Partial Differential Equation

QGASS Quantum Gradient-based Adaptive Stochastic Search

QND Quantum Non-Demolition

QRL Quantum Reinforcement Learning

QSTSO Quantum Spatio-Temporal Stochastic Optimization

QVO-MPC Quantum Variational Optimization - MPC

QVO-SS Quantum Variational Optimization - Single Shot

RDE Riccati Differential Equation

ReLU Rectified Linear Unit

RL Reinforcement Learning

RN Radon-Nikodym

RNN Recurrent Neural Network

ROM Reduced Order Model

SDE Stochastic Differential Equation

SGD Stochastic Gradient Descent

SME Stochastic Master Equation

SNN Sparse Neural Network

SOC Stochastic Optimal Control

SPDE Stochastic Partial Differential Equation

STDDP Spatio-Temporal DDP

STSO Spatio-Temporal Stochastic Optimization

SUMMARY

The majority of systems in nature have a spatio-temporal dependence and can be described by Partial Differential Equations (PDEs). They are ubiquitous in science and engineering, and are of rising interest among the control, robotics, and machine learning communities. Related methods usually treat these infinite dimensional problems in finite dimensions with reduced order models. This leads to committing to specific approximation schemes and the subsequent control laws cannot generalize outside of the approximation schemes. Additionally, related work does not consider spatio-temporal descriptions of noise that realistically represent the stochastic nature of physical systems. This thesis develops a variety of approaches for control optimization and co-design optimization for PDE and stochastic PDE (SPDE) systems from a unified perspective that can be applied to macroscopic systems in robotics and fluid dynamics, as well as microscopic systems in quantum mechanics. These approaches are each developed completely in the infinite dimensional Hilbert spaces where the systems are mathematically described, enabling the frameworks to be agnostic to the discretization scheme used to implement them. The first three developed approaches are applied in simulation to classical systems in fluid dynamics such as the Heat and Burgers equation. The fourth approach is developed for second-order SPDEs that arise in robotic systems, and is applied in simulation to systems in soft-robotics such as the Euler-Bernoulli equation and a biological model of a soft-robotic limb. Finally, several approaches are developed in the context of quantum feedback control of open quantum systems with non-demolition measurement, and one such approach is applied in simulation to perform explicit feedback control of the two qubit open quantum system.

CHAPTER 1
INTRODUCTION AND BACKGROUND

Systems that are among the most complex in physics and engineering are described by Partial Differential Equations (PDEs). PDEs are used to describe all the fundamental forces in nature, and are present in all fields of science and engineering. Often, in these complex natural processes, a variable such as temperature or displacement has values that are time varying on a spatial continuum. These systems are known as spatio-temporal processes and are ubiquitous in nature and engineering, including fields ranging from applied physics to robotics and autonomy [1, 2, 3, 4, 5, 6, 7, 8, 9].

The *Poisson-Vlassov* equation in plasma physics, the Heat, *Burgers* and *Navier-Stokes* equations in fluid mechanics, and the *Zakai* equation in classical filtering are just some examples of stochastic spatio temporal systems. Such systems are also found in a number of quantum processes, including numerous Stochastic Master Equations (SMEs) in quantum mechanics and the *Belavkin* SPDE in quantum filtering. Additionally, such systems are increasingly prevalent throughout the robotics community. Swarm robotics can be described by reaction-advection-diffusion PDEs [10]. Robot navigation in crowded environments can be described by Nagumo-like PDEs [11]. Soft robotic limbs can be modelled as damped Euler-Bernoulli systems [12]. The heat equation can be used for robotic motion planning [13] and has been shown to have equivalence to multi-agent consensus-based control laws for robot deployment problems [14].

These systems present extraordinary challenges from the perspective of control. Some of the major control-related challenges of spatio-temporal systems include dramatic under-actuation, high system dimensionality, and the design and/or placement of distributed actuators over a continuum of potential locations. These systems often have significant time delay from a control signal, and can have several bifurcations and multi-modal instabilities.

In addition, realistic representations of these systems are stochastic.

From the perspective of mathematics, the existence and uniqueness of solutions of SPDEs remains an open problem for many systems. When solutions do exist, they often have a weak notion of differentiability if at all. Furthermore, analysis of their dynamics must be treated with a suitable calculus over functionals. Finally their state vectors are often described by vectors in an infinte-dimensional time-indexed Hilbert space, even for scalar 1-dimensional SPDEs. Put together, mathematically consistent and numerically realizable algorithms for control of spatio-temporal systems represent many of the largest current-day challenges facing the robotics and automatic control communities.

The goal of this thesis is to derive and demonstrate control methodologies from a unified perspective that can be applied to macroscopic systems in robotics and fluid mechanics, as well as microscopic systems in quantum mechanics. The motivation behind the pursuit of control architectures for seemingly distant systems in separate disciplines is a system of unifying mathematics, and a common perspective that bridges foundational principles of Stochastic Optimal Control (SOC) theory and foundational principles of Information Theoretic Control (ITC), and is ultimately founded in the second law of thermodynamics.

Despite their ubiquity, their challenging nature has caused the theory of control of SPDEs to be introduced only in the last few decades [15, 16] and remains incomplete especially for stochastic boundary control. Numerical results and algorithms for distributed control of SPDEs are limited and typically require some model reduction approach [17, 18]. In [19], the authors approach the control of the stochastic Burgers equation through the Hamilton-Jacobi-Bellman (HJB) theory by applying the linear Feynman-Kac lemma; nevertheless, it lacks numerical results. In [20], the authors treat optimal control of linear deterministic PDEs by applying linear control theory, however this work is limited to linear PDEs. The book [16] gives a complete understanding of our ability so far, to apply optimal control theory to these systems.

Most notable among existing infinite dimensional control frameworks, [21] investigates

explicit solutions to the equation for the stochastic Burgers equation based on an exponential transformation, and [22] provides an extension of the large deviation theory to infinite dimensional spaces that creates connections to HJB theory. These and most other works on HJB theory for SPDEs mainly focus on theoretical contributions and leave literature with algorithms and numerical results tremendously sparse. Furthermore, HJB theory for boundary control has certain mathematical difficulties which impose limitations.

The majority of recent results are composed of a growing body of work that often rely on machine learning techniques, and seek control of PDEs by immediately reducing them to a set of Ordinary Differential Equations (ODEs) [23, 24, 25, 26, 27, 28]. They do not consider stochasticity and typically use standard tools from finite-dimensional control theory. In some cases, such as in [26], the resulting methods can violate stabilizability conditions, and in other cases, can lead to spillover instabilities [29, 30]. The majority of such approaches are focused around systems in fluid mechanics. In [27] the authors successfully control a Navier-Stokes system with reinforcement learning on policy networks in a deterministic, finite ODE setting. Similarly, [23] presents a Deep Recurrent Neural Network (RNN) framework with MPC to control a finite, deterministic ODE representation resulting from a Computational Fluid Dynamics (CFD) solver of a Navier-Stokes system.

In the soft robotics setting, [28] applies deep reinforcement learning, more precisely deep Q-learning, on a discrete finite markov decision process representation of a soft pneumatic-driven manipulator in order to obtain an open-loop position controller to control deflections at the tip. In [31], the authors similarly apply standard finite dimensional deep learning methods for policy and actuator co-design optimization of deformable body robots for locomotion by wrapping clustering and deep reinforcement learning around a differentiable simulator. Other recent finite-dimensional machine learning-based methods are covered in the review paper [32].

In the quantum setting, a large variety of methods have been applied in an open loop control setting, for closed quantum systems. Here we highlight a few methods, however

descriptions of such systems, along with a more complete review of recent work is provided in greater detail in chapter 7. A Pontryagin-based open loop control method is developed for two and three level quantum systems in [33]. Several open loop methods, such as the Krotov method and the conjugate gradient method are compared in [34] for control of coherence in three level systems. In [35], a Quantum Reinforcement Learning (QRL) is adopted to perform open loop control of a class of N-level quantum systems. While these methods can be developed in Hilbert spaces, they typically consider deterministic dynamics in a closed system setting.

Coupled to the challenges of control are the challenges of designing an effective actuation of the system such that the system experiences, and *maximizes*, the effect of some control policy. Such an actuation design problem is its own NP-hard problem due to the continua of possible actuator designs and possible placements over the spatial continuum of the domain of the SPDEs. Furthermore, actuation design performance is coupled to the performance of the control policy, and it is quite easy to confuse poor control performance with poor actuator design and vice versa. As a result, the challenge of a-priori deducing optimal actuation by a "human expert" that leverages the dynamics, even for relatively simple SPDEs, is quite daunting and often results in naive choices.

In the context of actuator co-design optimization for PDEs, several works have addressed optimal placement of actuators and sensors in the linear regime. In [36], minimum-norm control methods are used to place actuators for the stochastic heat equation. Similarly, H_∞ and H_2 objectives are used for placement of actuators in flexible structures in [37, 38, 39], and for the linearized Ginzburg-Landau equation in [40, 41]. Other methods leverage properties inherent to linear systems, such as symmetry properties in linear PDEs in [42], linear system Gramians, as in [43, 44], and level set methods based on Gramians that promise scalability in [45]. Aside from these methods which are appealing, yet constrained to linear systems, optimal actuator and sensor placement for stabilization of the nonlinear Kuramoto-Sivashinsky equation is demonstrated in [46]. They produce appealing results, however they

impose strong simplifying assumptions which limit their dimensionality. Finally, conditions for the existence of optimal actuator and sensor placement for semilinear PDEs are obtained in [47].

Finite dimensional methods generally rely on standard optimality principles from the finite dimensional SOC literature, namely the Dynamic Programming (or Bellman) principle and the stochastic Pontryagin Maximum principle [48, 49, 50], which typically provide solutions to the HJB equation that suffer from the curse of dimensionality. In contrast to typical Pontryagin and HJB methods, the Stochastic Differential Equation (SDE) control literature presents probabilistic representations of the HJB PDE that can solve scalability via sampling techniques [51, 52] including iterative sampling and/or parallelizable implementations [53].

In contrast, the SDE control literature presents probabilistic representations of the HJB PDE that can solve scalability via sampling techniques [51, 52] including iterative sampling and/or parallelizable implementations [54, 53]. Both forward-backward methods and sampling-based methods will be explored in this thesis, however an emphasis will be placed on sampling-based methods.

The foundation of these methodologies is a general principle stemming from statistical physics and thermodynamics, which has been shown to have applicability in SOC [55]:

$$\text{Free Energy} \leq \text{Work} - \text{Temperature} \times \text{Entropy} \tag{1.1}$$

This relation is an instantiation of the second law of thermodynamics, and optimization of this relation from a measure theoretic perspective gives rise to the well known Gibbs measure which is used in variational inference problems [56].

Connections between eq. (1.1) and the HJB equation were originally shown in finite dimensions and recently extended to infinite dimensions [57]. This connection is a primary motivator for their application to control, and highlights a set of differing perspectives on decision making under uncertainty that overlap for fairly general classes of stochastic

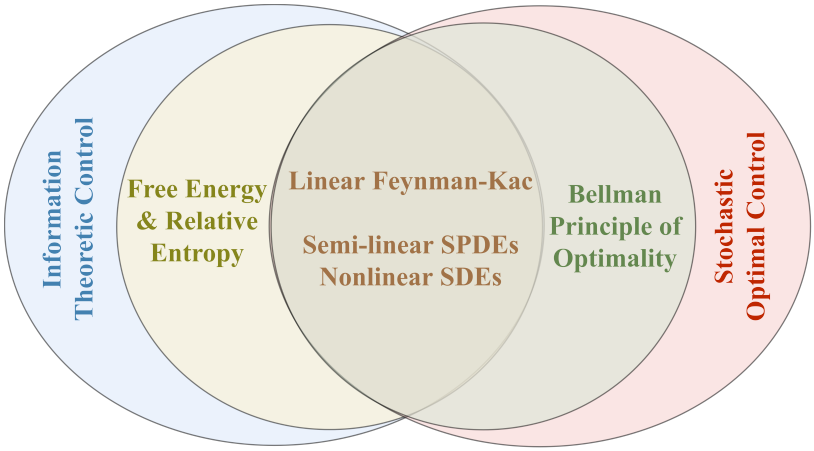


Figure 1.1: Connection between the free energy-relative entropy approach and stochastic Bellman Principle of Optimality.

systems, as depicted in Figure 1.1.

Through this lens, we approach a variety of control problems in a variety of disciplines, from robotics to fluid mechanics, to quantum mechanics. We unify these problems through a common mathematical description of systems that evolve in space and in time, and a common frame of reference from which we can derive optimization methodologies that result in both forward-backward SOC schemes and sampling-based ITC schemes.

Furthermore, as opposed to recent works which first require developing Reduced Order Models (ROMs) and then using standard approaches from Reinforcement Learning (RL) or MPC, we treat the SPDE system directly in Hilbert spaces and derive novel optimization methods for control of SPDEs directly. This set of approaches generally follow the path highlighted in red in Figure 1.2.

The primary advantages of performing optimization methods in Hilbert spaces is that the resulting algorithms are completely agnostic to the scheme used to discretize the PDE, which must after all must be discretized for simulation with discrete computation. Additionally, these methods enable our algorithms to simultaneously address control through actuators distributed throughout the spatial extent of the system, and actuators located on the boundary of the region.

This thesis is generally split into two main parts. Part I is dedicated to the optimization

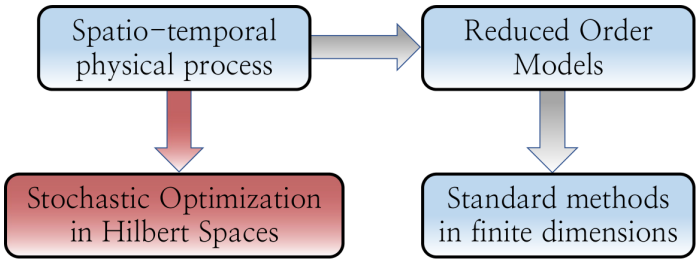


Figure 1.2: Optimization in Hilbert Spaces vs Optimization in finite dimensions
Our proposed approach versus traditional approaches.

algorithms for real-valued, macroscopic spatio-temporal systems. Here we develop optimization schema that deal with control optimization on deterministic PDEs from the SOC perspective, control optimization for SPDEs from the ITC perspective, and joint control and co-design optimization for SPDEs. Part II, on the other hand, deals with microscopic systems that live in complex-valued Hilbert spaces, and are governed by quantum mechanics. Here we primarily focus on schema that perform control optimization for SPDEs from the ITC perspective.

We start off part I by providing some mathematical preliminaries for both deterministic and stochastic spatio-temporal systems in chapter 2. These can be described as dynamics in the fields representation with functions acting on functions, and described in the time-indexed Hilbert space representation with operators acting on infinite dimensional vectors. This thesis does not provide the lengthy basic mathematical preliminaries on the calculus of functionals or on calculus in Hilbert spaces, however the interested reader can find these in [58] for functional calculus, and [2, 16] for stochastic calculus in infinite dimensions.

In chapter 3, we address the problem of optimal control of PDEs through a traditional SOC frame of reference. We work directly with the HJB equation, and derive a local linearization based forward-backward scheme for spatio-temporal systems, denoted Spatio-Temporal Differential Dynamic Programming. We demonstrate that the resulting forward-backward scheme is quite general, and can reproduce the standard DDP scheme in finite dimensions, as well as the standard LQR scheme in infinite dimensions. We further analyze convergence characteristics of the resulting forward-backward system. We highlight a

number of common discretization schemes that can handle the numerically stiff backward system, and we demonstrate the algorithm on two simulated PDE systems [59].

In chapter 4, we leverage the connection between SOC and ITC, and derive a variational optimization framework on time-indexed Hilbert spaces with an infinite dimensional stochastic calculus on Hilbert spaces. This approach mirrors similar methods that have been successfully applied to finite dimensional systems [60, 61]. The resulting control can be applied in either open-loop or MPC modes with either distributed actuators or boundary actuators, and is demonstrated in simulation for several semilinear SPDE systems in fluid dynamics [62].

The perspective explored in chapter 4 also enables us to seek a middle ground between recent results in Deep Learning (DL) and traditional SOC. In chapter 5, we approach SPDEs with infinite dimensional stochastic calculus, yet apply highly successful DL techniques to optimize the resulting measure-theoretic loss function. We develop a new method fusing together variational optimization, episodic reinforcement learning, and measure theoretic stochastic calculus in infinite dimensions. This results in an explicit closed-loop control scheme that in essense leverages the inherent stochasticity of the system for exploration in the space of policies, as demonstrated by application to several simulated semilinear SPDE systems in fluid dynamics [63].

In chapter 6, the approach in chapter 5 is further developed. The approach is framed as a control problem instead of a reinforcement learning problem, and we consider a problem were we concurrently perform optimization of the control policy as well as the design of the actuation of the system, which is referred to as control and co-design optimization. Namely, we wish to iteratively optimize both the design of the system actuation, and the signal sent to the actuators for control. Mathematical tools for importance sampling are extended to second-order SPDEs and the resulting approach is applied to optimal control and co-design of numerous SPDE systems in fluid dynamics and robotics. These include a simplified linear model of a stochastic soft-robotic system, and more interestingly a detailed and complex

nonlinear, 2D, 2^{nd}-order stochastic model of soft-robotic limbs with origins in biological dynamic modeling of the appendages of the octopus vulgaris [64].

Next, we turn our attention to stochastic control of quantum systems in part II. In chapter 7, we introduce the currently dominant paradigm in quantum control. We introduce the mathematical description of open quantum systems and describe their dynamics. We also introduce the notion of quantum non-demolition measurement, and arrive at the Belavkin equation or stochastic master equation. Finally we develop the optimal control problem based on the stochastic master equation, and arrive at the HJB equation for open quantum systems conditioned on a weak quantum non-demolition measurement. This thesis does not provide the lengthy basic mathematical preliminaries on quantum mechanics, however the interested reader can find these in [65] for an introduction to quantum mechanics, [66] for quantum measurement, and [67] or the cited works by V.P. Belavkin for quantum stochastic calculus.

Based on the connections between SOC and ITC in infinite dimensions, in chapter 8 we derive an associated change of drift for open quantum systems based on the notion that the diffusion process is a classical Wiener process. This is then applied to a quantum variational optimization problem in an approach anaologous to chapter 4 for open quantum systems in order to develop a quantum feedback control architecture, where explicit feedback appears due to the form of the Radon-Nikodym derivative. We attempt to apply the resulting update scheme to two popular open quantum system experiments, and observe the shortcomings of the approach.

The aformentioned shortcomings motivate an alternate method for feedback control of open quantum systems which does not require an importance sampling step. In chapter 9, we develop such an approach based on the theory of stochastic approximation, which ultimately has many similarities to the update scheme in chapter 4. Several control schema are proposed, which yield a variety of potential architectures, and one is selected for demonstration in a simulated experiment of a two qubit quantum system. We demonstrate that the so-

called quantum gradient-based adaptive stochastic search framework can effectively train a feedback policy network to stabilize one of the two qubit states of maximal entanglement, or Bell states, and more importantly, can outperform a landmark approach for feedback control of open quantum systems.

Motivated by the notion of dynamic compensation for feedback control, we return to the original ITC framework developed in chapter 8. We prove a Girsanov theorem for a new change of drifts, and develop a change of measures, or Radon-Nikodym derivative, which does not require inversion, in contrast to the change of measures obtained in chapter 8. Based on this result, we are able to develop control optimization schema for open loop policies and MPC policies akin to chapter 4, and explicit feedback policies akin to chapter 5. The resulting approaches are general, and can be applied to virtually *any* semilinear open quantum system experiment conditioned on weak non-demolition measurement. Finally, the thesis is concluded in chapter 11.

Part I

Control Optimization for Spatio-Temporal Systems in Robotics and Fluid Mechanics

CHAPTER 2

MATHEMATICAL PRELIMINARIES

2.1 Spatio-Temporal Systems in Fields and Hilbert Space Representations

Let $D \subseteq \mathbb{R}^n$ denote a measurable connected open domain of $\mathbb{R}^n$ describing the space on which the system evolves. Let $S \subseteq \mathbb{R}^n$ denote the boundary of D, let $\bar{D}$ denote the closure of the domain, i.e. $\bar{D} = D \cup S$, and let $T = [t_0, t_f]$ denote some arbitrary time domain. In fields representation, a general form of a deterministic PDE dynamical system is given by

$$\partial_t X(t,x) = F(t,x,X(t,x),U_d(t,x)), \quad x \in D \tag{2.1}$$

$$0 = N(t,x,X(t,x),U_b(t,x)), \quad x \in S \tag{2.2}$$

$$X(t_0,x) = X_0(x), \quad x \in \bar{D}, \tag{2.3}$$

where $X : T \times D \to \mathbb{R}^n$ is the state. This problem has two measurable control functions, $U_b : T \times S \to \mathbb{R}^l$ which correspond to actuation on the boundary, and $U_d : T \times D \to \mathbb{R}^k$ which corresponds to actuation distributed throughout the field excluding the boundary. The dynamics evolve by some measurable functional $F : T \times D \times \mathbb{R}^n \times \mathbb{R}^k \to \mathbb{R}^n$ that is potentially nonlinear in the state function $X(t,x)$ or the control function $U_d(t,x)$, with a boundary condition functional $N : T \times S \times \mathbb{R}^n \times \mathbb{R}^l \to \mathbb{R}^n$ that is also potentially a nonlinear functional of the state or control functions, and can be any type of boundary condition (e.g. Neumann, Dirichlet, etc.). In this notation, the system evolution is described by finite dimensional vector-valued functionals with spatio-temporal functions as arguments.

We can equivalently write eqs. (2.1) to (2.3) in the time-indexed Hilbert spaces perspective by first properly defining Hilbert spaces, as in [68]. Let $L_2^n(D)$ denote the Hilbert space

of n-vector functions square integrable over D with inner product

$$\left\langle X_1, X_2 \right\rangle = \int_D X_1^\top(x) X_2(x) \mathrm{d}x, \tag{2.4}$$

where $\mathrm{d}x = \mathrm{d}x_1 \mathrm{d}x_2 \cdots \mathrm{d}x_n$ is shorthand notation for the generalized volume integration over $\mathbb{R}^n$. This is the Hilbert space of the domain, and we similarly define the Hilbert space over the boundary. Let $L_2^n(S)$ denote the Hilbert space of n-vector functions square integrable over S with inner product

$$\left\langle X_1, X_2 \right\rangle_S = \int_S X_1(\xi) X_2(\xi) \mathrm{d}S_\xi, \tag{2.5}$$

where $\mathrm{d}S_\xi$ is an infinitesimal surface element of the boundary at a point $\xi \in S$. Let $\mathscr{L}(U,V)$ denote the space of linear bounded operators from U into V.

If we regard $X(t,x)$ as an element of $L_2^n(D)$, then we can rewrite eqs. (2.1) to (2.3) as

$$\frac{\mathrm{d}}{\mathrm{d}t} X(t) = F\big(t, X(t), U_d(t)\big), \quad X \in L_2^n(D), \ t \in T \tag{2.6}$$

$$0 = N\big(t, X(t), U_b(t)\big), \quad X \in L_2^n(S), \ t \in T \tag{2.7}$$

$$X(t_0) = X_0, \tag{2.8}$$

where $X(t), X_0 \in L_2^n(D)$ are respectively the Hilbert space state vector and initial conditions, $U_d(t) \in L_2^k(D)$ is the Hilbert space distributed control vector, $U_b \in L_2^l(S)$, is the Hilbert space boundary control vector, $F : T \times L_2^n(D) \times L_2^k(D) \to L_2^n(D)$ is a potentially nonlinear measurable function on the domain Hilbert space, and $N : T \times L_2^n(D) \times L_2^l(S) \to L_2^n(S)$ is a potentially nonlinear measurable function on the boundary Hilbert space. In this notation, the system evolution is described by infinite dimensional operator functions acting on time-indexed infinite dimensional vectors on Hilbert spaces of square integrable functions.

The key difference of these two perspectives is in the representation of the spatial continuum of the domain. In the former, the spatial continuum is represented explicitly as

spatial dependence of the state, whereas in the latter, the state vector lives on a space of functions to describe its spatial dependence. Note that the field functional perspective of eqs. (2.1) to (2.3) and the time-indexed Hilbert space perspective of eqs. (2.6) to (2.8) are consistent in the sense that they share identical solutions up to the transformation between the perspectives used above. This transformation can be described as 'lifting' the system into infinite dimensional Hilbert spaces.

Assumption 2.1. *The PDE system in fields representation given by eqs.* (2.1) *to* (2.3) *is well posed in the sense of Hadamard, and admits a unique weak solution* $X(t,x)$, $t \in T, x \in \bar{D}$ *for each initial condition* $X_0(x) \in \mathbb{R}^n$.

Depending on the specific form of the PDE, this assumption can have varying degrees of severity, however in general it is a mild assumption. Please refer to [69] for more details on existence and uniqueness of various PDEs. Despite the potential severity, it is an assumption that is required henceforth. Note also that if assumption 2.1 holds, then the PDE system in Hilbert space representation given by eqs. (2.6) to (2.8) is also well posed in the sense of Hadamard, and admits a unique weak Hilbert space solution $X(t) \in L_2^n(\bar{D})$, $t \in T$ for each Hilbert space initial condition $X_0 \in L_2^n(\bar{D})$. The two notational perspectives describe the *same* system, and this remark simply states that if the the former has unique weak solutions, then so does the latter.

Throughout chapter 3, we go back and forth between these two notational perspectives: the spatially varying fields perspective, and the time-indexed Hilbert space perspective. While the fields perspective demonstrates the spatial integration that is central to the Volterra-Taylor expansions more clearly, the time-indexed Hilbert space perspective will often yield a more compact notation that is easier to treat with familiar algebraic operations. Whenever we suppress the dependencies on the spatial variable x, the variables are assumed to be in time-indexed Hilbert spaces.

2.2 Stochastic Spatio-Temporal Systems in Hilbert Spaces

Common to all of the stochastic approaches considered here is a time-indexed Hilbert space perspective of stochastic systems evolving over space and time. Let $D \subseteq \mathbb{R}^n$ denote a connected open domain of $\mathbb{R}^n$ describing the space on which the system evolves, and $S \subseteq \mathbb{R}^n$ the boundary of D. The closure of D is denoted $\bar{D} = D \cup S$. Also, let $(\Omega, \mathscr{F}, \mathbb{P})$ be a probability space with filtration $\mathscr{F}_t$, $t \in [0,T]$. A natural way of describing a general SPDE is the fields representation, given by

$$\partial_t X(t,x) = F(t,x,X(t,x),U_d(t,x)) + \frac{1}{\rho} G\big(t,\xi,X(t,x)\big)\partial_t W(t), \quad x \in D \tag{2.9}$$

$$0 = N(t,\xi,X(t,\xi),U_b(t,\xi)), \quad \xi \in S \tag{2.10}$$

$$X(t_0,x) = X_0(x), \quad x \in \bar{D} \tag{2.11}$$

where $X : [0,T] \times D \to \mathbb{R}^n$ is the state. This problem has two control functionals, $U_b : [0,T] \times S \to \mathbb{R}^l$ which correspond to actuation on the boundary, and $U_d : [0,T] \times D \to \mathbb{R}^k$ which corresponds to actuation distributed throughout the field excluding the boundary. The dynamics evolve by some functional $F : [0,T] \times D \times \mathbb{R}^n \times \mathbb{R}^k \to \mathbb{R}^n$ that is potentially nonlinear in the state function $X(t,x)$ or the control function $U_d(t,x)$, with a boundary condition functional $N : [0,T] \times S \times \mathbb{R}^n \times \mathbb{R}^l$ that is also potentially a nonlinear functional of the state or control functions, and can be any type of boundary condition (e.g. Neumann, Dirichlet, etc.).

It is often useful to describe this system as evolving on time-indexed, infinite dimensional Hilbert spaces. Let H and U be separable Hilbert spaces. A subset of the systems described by eqs. (2.9) and (2.10) can be described in the following *semi-linear* controlled and

uncontrolled form [16]

$$\mathscr{L}: \quad \mathrm{d}X = \mathscr{A}X\mathrm{d}t + \hat{F}(t,X)\mathrm{d}t + \frac{1}{\sqrt{\rho}}G(t,X)\mathrm{d}W(t), \tag{2.12}$$

$$\mathscr{L}^{(i)}: \quad \mathrm{d}X = \mathscr{A}X\mathrm{d}t + \hat{F}(t,X)\mathrm{d}t + G(t,X)\left(\mathscr{U}^{(i)}(t,X;\theta)\mathrm{d}t + \frac{1}{\sqrt{\rho}}\mathrm{d}W(t)\right), \tag{2.13}$$

where $X(0)$ is an $\mathscr{F}_0-$measurable, $H-$valued random variable, and $\mathscr{A}: D(\mathscr{A}) \subset H \to H$ is a linear operator, where $D(\mathscr{A})$ denotes here the domain of $\mathscr{A}$. $\hat{F}: \mathbb{R}\times H \to H$ and $G: \mathbb{R}\times U \to H$ are nonlinear operators that satisfy properly formulated Lipschitz conditions associated with the existence and uniqueness of solutions to eq. (2.12) as described in [70, Theorem 7.2]. The term $\mathscr{U}^{(i)}(t,X;\theta)$ will differ between approaches, but represents some actuation or control action onto the system. We view these dynamics in an iterative fashion in order to realize an iterative method. As such, the superscript (i) refers to the iteration number.

The term $W(t) \in U$ corresponds to a *Hilbert space Wiener process*, which is a generalization of the Wiener process in finite dimensions. When this noise profile is spatially uncorrelated, we call it a *cylindrical Wiener process*, which requires the added assumptions on $\mathscr{A}$ in [2, Hypthesis 7.2] in order to form a contractive, unitary, linear semigroup, which is required to guarantee existence and uniqueness of $\mathscr{F}_t$-adapted weak solutions $X(t), t \geq 0$. A thorough description of the Wiener process in Hilbert spaces, along with its various forms can be found in Appendix A. For generality, eqs. (2.12) and (2.13) introduce the parameter $\rho \in \mathbb{R}$, which acts as a uniform scaling of the covariance of the Hilbert space Wiener process.

We denote $\langle\cdot,\cdot\rangle_D$ as the inner product in a Hilbert space D, and $C([0,T];H)$ the space of continuous processes in H for $t \in [0,T]$. We sometimes suppress the subscript of the inner product for simplicity of notation when the space of the inner product is otherwise clear. Define the measure on the path space of uncontrolled trajectories produced by eq. (2.12) as $\mathscr{L}$ and define the measure on the path space of controlled trajectories produced by eq. (2.13)

Table 2.1: Examples of commonly known semi-linear PDEs in a *fields representation* with subscript x representing partial derivative with respect to spatial dimensions and subscript t representing partial derivatives with respect to time. The associated operators $\mathscr{A}$ and $F(t,X)$ in the Hilbert space formulation are colored blue and violet, respectively.

Equation Name	**Partial Differential Equation**	**Field State**
Heat	$u_t = \varepsilon u_{xx}$	Heat/temperature
Burgers (viscous)	$u_t = \varepsilon u_{xx} - u u_x$	Velocity
Nagumo	$u_t = \varepsilon u_{xx} + u(1-u)(u-\alpha)$	Voltage
Allen-Cahn	$u_t = \varepsilon u_{xx} + u - u^3$	Phase of a material
Navier-Stokes	$u_t = \varepsilon \Delta u - \nabla p - (u \cdot \nabla) u$	Velocity
Nonlinear Schrodinger	$u_t = \frac{1}{2} i u_{xx} + i\lvert u\rvert^2 u = 0$	Wavefunction
Korteweg-de Vries	$u_t = -u_{xxxx} - 6 u u_x$	Plasma wave
Kuramoto-Sivashinsky	$u_t = -u_{xx} - u_{xxxx} - u u_x$	Flame front

as $\mathscr{L}^{(i)}$. With these measures, we use the notation $\mathbb{E}_{\mathscr{L}}$ to denote expectations over paths as Feynman path integrals.

Many physical and engineering systems can be written in the abstract form of eq. (2.12) by properly defining operators $\mathscr{A}$, F and G along with their corresponding domains. Examples can be found in our simulated experiments, as well as table 2.1, with more complete descriptions in [70, Chapter 13]. The goal of this thesis is to establish control methodologies for deterministic and stochastic versions of such systems.

CHAPTER 3

SPATIO-TEMPORAL DIFFERENTIAL DYNAMIC PROGRAMMING FOR CONTROL OF FIELDS

Infinite-dimensional methods found in the control theory literature [71, 72] often perform control via linear or linearization-based approaches, which include LQR approaches for linear PDEs, and forward-backward approaches, which include approaches due to the Pontryagin Maximum Principle (PMP) [72, 73, 74]. Indeed local linearization methods allow for optimal solutions of an approximate problem, however require knowledge of linearization points a-priori. On the other hand, forward-backward schemes provide a nominal trajectory and optimization-based control update scheme at the expense of the backpropagation of a coupled system of equations.

In contrast to Pontryagin methods which yield a state-independent backward equation and an open-loop controller, methods founded on the Bellman principle of optimality utilize backward equations that are state-dependent and yield closed-loop control solutions. Methods such as DDP have decades of established history in the finite dimensional automatic control literature. Modern variations include control limits [75], state constraints [76], receding horizons [77], belief space control [78, 79], game-theoretic control [80], control on Lie groups [81], and using polynomial chaos variational integrators [82].

A previous attempt exists to extend the DDP framework to spato-temporal systems in infinite dimensions [83], however this approach has several flaws and mathematical inconsistencies, as pointed out in [68]. Additionally, the DDP method has had significant growth since the early works [84]. Decades of advancement include linearization around the nominal trajectory as opposed to the optimal trajectory which decreases sensitivities of convergence behavior to the initial conditions, regularization in the second order backward equation to increase numerical stability, treatment of state and control constraints, and

optimization over time horizon.

In light of the apparent literature gap, this chapter is devoted to the development of DDP methods for spatio-temporal systems in infinite dimensions. Specifically, we derive the Spatio-Temporal DDP (STDDP) framework incorporating modern theoretical techniques, we demonstrate that the resulting system of forward-backward equations generalizes both the LQR solution in infinite dimensions and DDP in finite dimensions, we provide a proof of convergence for the resulting system of continuous-time forward-backward equations, we explore and develop numerical approaches to handle sensitivities that arise due to discretization, and apply the resulting algorithm to linear and nonlinear spatio-temporal PDE systems. In contrast to recent machine learning methods, our optimization is developed entirely in Hilbert spaces, and represents an *optimize-then-discretize* approach. As a result, the framework is a continuous-time formulation which is *agnostic* to discretization scheme during implementation.

3.1 Problem Statement

In order to arrive at the optimal control problem, we first define the measurable cost functional in fields representation as

$$J\big(t,X(t,x),U_d(t,x),U_b(t,x)\big) := \phi\big(t_f,X(t_f,x)\big) + \int_{t_0}^{t_f} L\big(t,X(t,x),U_d(t,x),U_b(t,x)\big)\mathrm{d}t, \tag{3.1}$$

where $\phi : T \times \mathbb{R}^n \to \mathbb{R}$ is some measurable real-valued terminal cost functional, and $L : T \times \mathbb{R}^n \times \mathbb{R}^k \times \mathbb{R}^l \to \mathbb{R}$ is a measurable real-valued running cost functional. In time-indexed Hilbert spaces, the cost functional becomes

$$J\big(t,X(t),U_d(t),U_b(t)\big) = \phi\big(t_f,X(t_f)\big) + \int_{t_0}^{t_f} L\big(t,X(t),U_d(t),U_b(t)\big)\mathrm{d}t, \tag{3.2}$$

where $J : T \times L_2^n(D) \times L_2^k(D) \times L_2^l(S) \to \mathbb{R}$, $\phi : T \times L_2^n(D) \to \mathbb{R}$, and $L : T \times L_2^n(D) \times L_2^k(D) \times L_2^l(S) \to \mathbb{R}$ are the equivalent measurable real-valued functionals in Hilbert spaces.

Define the continuous-time, finite-horizon, infinite dimensional control problem as

$$\inf_{U_d, U_b} J\big(t, X(t), U_d(t), U_b(t)\big) \tag{3.3}$$

subject to the Hilbert space dynamics in eqs. (2.6) to (2.8). One can define the so called value functional over this optimal control problem as

$$V\big(t, X(t)\big) := \inf_{U_d, U_b} \Big[J\big(t, X(t), U_d(t), U_b(t)\big) \Big]. \tag{3.4}$$

Due to the Bellman Principle of Optimality, one can additionally form the HJB equation [83, 68], which is given formally as

$$-\partial_t V\big(t, X(t)\big) = \inf_{U_d, U_b} \Big[L(t, X, U_d, U_b) + \Big\langle V_X(t, X), F(t, X, U_d) \Big\rangle \Big], \tag{3.5}$$

$$V\big(t_f, X(t_f)\big) = \phi\big(t_f, X(t_f)\big) =: V_f \in \mathbb{R}, \tag{3.6}$$

where we write $\partial_t = \frac{\partial}{\partial t}$ to denote the normal partial derivative of a function with respect to a variable, and use subscript X, U_b, or U_d to denote the Gateaux partial derivative of a functional or operator with respect to an operator function. One can carry out the same derivation using Volterra's notion of functional derivative [58]. Note that $V(X(t), t)$ is a function of time, and a functional of $X(t)$.

It is important to note that the HJB equation in eq. (3.5) is a backwards nonlinear PDE, which has no explicit dependence of the right hand side on the boundary control U_b in this initial formulation. Instead, in eq. (3.5) the boundary control only enters implicitly on the right hand side. This will be explored in greater detail in the subsequent derivation, wherein the Green's theorem must be applied in order to reveal an explicit dependence of the resulting HJB equation on the boundary control, after which a Newton style minimization is

performed to yield the optimal distributed and boundary control updates. This process starts with the following assumption.

Assumption 3.1. *The backwards PDE in eq.* (3.5) *admits a unique viscosity solution* $V\left(t,X(t)\right)$, $t \in T$, $X(t) \in L_2^n(\bar{D})$ *for each terminal condition* $V\left(X(t_f),t_f\right) = V_f := \phi\left(t_f,X(t_f)\right) \in \mathbb{R}$.

The DDP framework solves the HJB equation in eq. (3.5) iteratively via expansions of the value functional, cost functional, dynamics operator function, and boundary operator function to given order. Typically, the value functional and cost functional are expanded to second order so that the resulting HJB becomes a quadratic optimization problem with a unique optimal control minimizer.

Quadratic expansions also allow for proofs of global convergence and even proofs of quadratic convergence, that in finite dimensions, initially relied on well known convergence properties of the Newton method of optimization [85, 86] for quadratic problems. Under similar reasoning, the dynamics are typically either expanded to first or second order.

3.2 Expansions of the Cost, Value, Field, and Boundary

The approach in this paper is a spatio-temporal DDP approach that is analogous to the finite dimensional DDP apparoch of [87]. Therein, the authors discuss the fundamental differences between their derivation, and the original derivation by Jacobson and Mayne [84]. The derivation by Jacobson and Mayne, of which a similar flavor is followed in [83], is based on the restrictive assumption that the nominal control trajectory $\bar{u}$ is sufficiently close to the optimal control solution u^*. This is circumvented by performing expansions around a nominal trajectory. Define a nominal state and control triple $(\bar{X},\bar{U}_d,\bar{U}_b)$ and the variations $\delta X := X - \bar{X}$, $\delta U_d := U_d - \bar{U}_d$, and $\delta U_b := U_b - \bar{U}_b$. In order to properly write the expansions, we require the following assumption. The conditions needed to satisfy this assumption can be found in [52].

Assumption 3.2. *The dynamics function F and boundary function N are differentiable almost everywhere, the running cost functional L and terminal cost functional ϕ are twice differentiable almost everywhere, and the value functional V is three times differentiable almost everywhere. These stated derivatives are defined in the Gateaux sense with respect to the state and control triple (X, U_d, U_b), and are square integrable in the Lebesgue sense. That is, the stated Gateaux derivative of each functional exists $\forall (X, U_d, U_b) \in \left(L_2^n(D), L_2^k(D), L_2^l(S)\right)$ except on a properly defined set of measure zero.*

As previously stated, the value functional is a function of time t, but a functional of the spacetime function $X(t,x)$. Thus the value functional is expanded via a Volterra-Taylor functional expansion [58]

$$\begin{aligned} V\left(t, \bar{X}(t,x) + \delta X(t,x)\right) = V\left(t, \bar{X}(t,x)\right) + \int_D V_X^\top \left(t, \bar{X}(t,x)\right) \delta X(t,x) \mathrm{d}x \\ + \frac{1}{2} \int_D \int_D \delta X^\top (t,x) V_{XX}(t,x,y) \delta X(t,y) \mathrm{d}x \mathrm{d}y + O(\delta^3). \end{aligned} \tag{3.7}$$

We maintain connection to the Hilbert space perspective by defining Hilbert space operators for each kernel function. Define the operator $V_{XX}(t,X) \in \mathscr{L}\left(L_2^n(D), L_2^n(D)\right)$ as

$$V_{XX}\left(t,X\right) W(t) := \int_D V_{XX}(t,x,y) W(t,y) \mathrm{d}y, \tag{3.8}$$

where $V_{XX}(t,x,y)$ is the kernel function. In order to form the left-hand side of the HJB eq. (3.5), we apply a re-arranged definition of the total differential [58], given by

$$\partial_t(\cdot) = \frac{\mathrm{d}}{\mathrm{d}t}(\cdot) - \left\langle (\cdot)_X, F(t,X,U_d) \right\rangle, \tag{3.9}$$

which holds for any functional that explicitly depends on X and t. In order to simplify notation, we suppress arguments when functionals are evaluated on the nominal trajectory triple. We apply eq. (3.9) to each term on the right-hand side of eq. (3.7), to yield the

left-hand side of the HJB, which in Hilbert spaces, has the form

$$-\partial_t V(t,\bar{X}+\delta X) = -\frac{\mathrm{d}}{\mathrm{d}t}\left(V + \left\langle V_X, \delta X\right\rangle + \frac{1}{2}\left\langle \delta X, V_{XX}\delta X\right\rangle\right) + \left\langle V_X, F\right\rangle + \left\langle V_{XX}F, \delta X\right\rangle + \frac{1}{2}\left\langle FV_{XXX}\delta X', \delta X\right\rangle, \tag{3.10}$$

where for the third order Gateaux derivative V_{XXX}, we have defined the tensor operator in time-indexed Hilbert spaces $V_{XXX}(t,X) \in \mathscr{L}\left(L_2^n(D)\times L_2^n(D), L_2^n(D)\right)$ as

$$U(t)V_{XXX}\big(t,X(t)\big)W(t) := \int_D\int_D U^\top(t,x)V_{XXX}(t,x,y,z)W(t,y)\mathrm{d}x\mathrm{d}y. \tag{3.11}$$

The 4-D kernel function $V_{XXX}(t,x,y,z)$ is assumed to be symmetric about all three spatial axes for simplicity.

Next, we expand the cost functional with a Volterra-Taylor expansion to second order, which in time-indexed Hilbert spaces has the form

$$\begin{aligned} L(t,\bar{X}+\delta X,\bar{U}_d+\delta U_d,\bar{U}_b+\delta U_b) = L &+ \left\langle L_X,\, \delta X\right\rangle + \left\langle L_{U_d}, \delta U_d\right\rangle + \left\langle L_{U_b},\, \delta U_b\right\rangle_S \\ &+ \frac{1}{2}\left\langle \delta X, L_{XX}\delta X'\right\rangle + \frac{1}{2}\left\langle \delta U_d, \left(L_{U_dX} + L_{XU_d}^\top\right)\delta X'\right\rangle \\ &+ \frac{1}{2}\left\langle \delta U_b, \left(L_{U_bX} + L_{XU_b}^\top\right)\delta X'\right\rangle_S \\ &+ \frac{1}{2}\left\langle \delta U_d, L_{U_dU_d}\delta U_d'\right\rangle + \frac{1}{2}\left\langle \delta U_b, L_{U_bU_b}\delta U_b'\right\rangle_S \\ &+ O(\delta^3), \end{aligned} \tag{3.12}$$

where we have defined the operators

$$
\begin{aligned}
L_{XX}\big(t,X(t),U_b(t),U_d(t)\big)W(t) &:= \int_D L_{XX}(t,x,y)W(t,y)\mathrm{d}y \\
L_{XU_d}\big(t,X(t),U_b(t),U_d(t)\big)W(t) &:= \int_D L_{XU_d}(t,x,y)W(t,y)\mathrm{d}y \\
L_{XU_b}\big(t,X(t),U_b(t),U_d(t)\big)W(t) &:= \int_S L_{XU_b}(t,\xi,\eta)W(t,\eta)\mathrm{d}S_\eta \\
L_{U_dU_d}\big(t,X(t),U_b(t),U_d(t)\big)W(t) &:= \int_D L_{U_dU_d}(t,x,y)W(t,y)\mathrm{d}y \\
L_{U_bU_b}\big(t,X(t),U_b(t),U_d(t)\big)W(t) &:= \int_S L_{U_bU_b}(t,\xi,\eta)W(t,\eta)\mathrm{d}S_\eta,
\end{aligned}
$$

and similarly defined operators for L_{U_dX}, L_{U_bX}.

Assumption 3.3. *The measurable kernel functions L_{XX}, L_{XU_d}, L_{XU_b} are spatially symmetric and positive semi-definite. The measurable kernel functions V_{XX}, $L_{U_dU_d}$, $L_{U_bU_b}$ are spatially symmetric and positive definite. The omitted cross term operators $L_{U_bU_d}$ and $L_{U_dU_b}$ are null operators.*

Note the assumption that cross terms between boundary and distributed control (i.e. $L_{U_bU_d}$ and $L_{U_dU_b}$) are zero. This is a fairly benign assumption since cost functionals are often composed of pure quadratics in either U_d or U_b, but not both. Including these cross terms also yields optimal update equations for boundary and distributed control that are coupled to each other, and thus impose mathematical and implementation difficulties.

Next, the dynamics and boundary are expanded around the nominal trajectory. The dynamics functional $F\big(t,X(t),U_d(t)\big)$ and boundary functional $N\big(t,X(t),U_b(t)\big)$ map into $L_2^n(D)$ and $L_2^n(S)$, respectively, and are not real-valued functionals, so it is appropriate to treat them as operator functions instead of as functionals despite having explicit dependence on functions $\bar{X},\bar{U}_d,\bar{U}_b$. In Hilbert space notation, the operator Taylor expansion of the

dynamics and boundary have the form

$$F(t,\bar{X}+\delta X,\bar{U}_d+\delta U_d) = F(t,\bar{X},\bar{U}_d)+F_X^\top(t,\bar{X},\bar{U}_d)\delta X + F_{U_d}^\top(t,\bar{X},\bar{U}_d)\delta U_d + O(\delta^2), \tag{3.13}$$

$$N(t,\bar{X}+\delta X,\bar{U}_b+\delta U_b) = N(t,\bar{X},\bar{U}_b)+N_X^\top(t,\bar{X},\bar{U}_b)\delta X + N_{U_b}^\top(t,\bar{X},\bar{U}_b)\delta U_b + O(\delta^2), \tag{3.14}$$

where transposes denote the associated transpose operators. We obtain the right-hand side of the HJB eq. (3.5) by plugging eqs. (3.12) to (3.14), and a Volterra-Taylor expansion of V_X. After simplification, the right-hand side of the HJB eq. (3.5) becomes

$$\begin{aligned}
\inf_{\delta U_d,\delta U_b}\Bigg[& L + \Big\langle L_X,\, \delta X\Big\rangle + \Big\langle L_{U_d},\delta U_d\Big\rangle + \Big\langle L_{U_b},\,\delta U_b\Big\rangle_S + \frac{1}{2}\Big\langle \delta X, L_{XX}\,\delta X'\Big\rangle \\
& + \frac{1}{2}\Big\langle \delta U_d,\, \Big(L_{U_dX}+L_{XU_d}^\top\Big)\delta X\Big\rangle + \frac{1}{2}\Big\langle \delta U_b,\, \Big(L_{U_bX}+L_{XU_b}^\top\Big)\delta X'\Big\rangle_S \\
& + \frac{1}{2}\Big\langle \delta U_d, L_{U_dU_d}\,\delta U_d'\Big\rangle + \frac{1}{2}\Big\langle \delta U_b, L_{U_bU_b}\,\delta U_b'\Big\rangle_S + \Big\langle V_X,\, F\Big\rangle + \Big\langle V_X,\, F_X^\top\delta X\Big\rangle \\
& + \Big\langle V_X,\, F_{U_d}^\top\delta U_d\Big\rangle + \Big\langle \delta X,\, V_{XX}F\Big\rangle + \Big\langle \delta X,\, V_{XX}F_X^\top\delta X'\Big\rangle \\
& + \Big\langle \delta X,\, V_{XX}F_{U_d}^\top\delta U_d\Big\rangle + \frac{1}{2}\Big\langle \delta X,\, V_{XXX}F\,\delta X'\Big\rangle\Bigg].
\end{aligned} \tag{3.15}$$

Equating eq. (3.10) to eq. (3.15) and canceling common terms yields

$$\begin{aligned}
& -\frac{\mathrm{d}}{\mathrm{d}t}\Big(V+\Big\langle V_X,\delta X\Big\rangle+\frac{1}{2}\Big\langle \delta X, V_{XX}\delta X'\Big\rangle\Big) \\
& = \inf_{\delta U_d,\delta U_b}\Bigg[L + \Big\langle L_X,\, \delta X\Big\rangle + \Big\langle L_{U_d},\delta U_d\Big\rangle + \Big\langle L_{U_b},\,\delta U_b\Big\rangle_S + \frac{1}{2}\Big\langle \delta X, L_{XX}\,\delta X'\Big\rangle \\
& \qquad + \frac{1}{2}\Big\langle \delta U_d,\, \Big(L_{U_dX}+L_{XU_d}^\top\Big)\delta X\Big\rangle + \frac{1}{2}\Big\langle \delta U_b,\, \Big(L_{U_bX}+L_{XU_b}^\top\Big)\delta X'\Big\rangle_S \\
& \qquad + \frac{1}{2}\Big\langle \delta U_d, L_{U_dU_d}\,\delta U_d'\Big\rangle + \frac{1}{2}\Big\langle \delta U_b, L_{U_bU_b}\,\delta U_b'\Big\rangle_S + \Big\langle V_X,\, F_X^\top\delta X\Big\rangle \\
& \qquad + \Big\langle V_X,\, F_{U_d}^\top\delta U_d\Big\rangle + \Big\langle \delta X,\, V_{XX}F_X^\top\delta X'\Big\rangle + \Big\langle \delta X,\, V_{XX}F_{U_d}^\top\delta U_d\Big\rangle\Bigg].
\end{aligned} \tag{3.16}$$

Remark 3.1. *The exact singleton Newton minimizer δU_b^* of the approximate HJB equation*

eq. (3.16) *does not incorporate the value functional* $V(t,\bar{X})$ *or its derivatives* $V_X(t,\bar{X})$, $V_{XX}(t,\bar{X})$.

This is an important point. The value functional is defined as the minimization surface of the original problem in eq. (3.4) and the apparent decoupling between the optimal update δU_b^* and the value functional and/or its derivatives within the resulting approximate HJB eq. (3.16) yields a naive update. The authors in [83] and [88] realize this fact, and use the Green's theorem in order to incorporate boundary information into specific terms in eq. (3.16). However, there are errors in their application of Green's theorem in the multivariate case, as noted in [68].

3.3 Green's Theorem in Hilbert Spaces

Green's theorem is used widely in calculus to relate the volume integral of the interior of a region to a surface integral of its boundary. In the context of STDDP, it allows us to capture pertinent effects of the value function on the boundary.

Assumption 3.4. *F_X is a linear operator with standard form given by $A_x(t,x)$ in [68], and N_X is a linear operator with standard form given by $\beta_A(t,\xi)$ in [68].*

Theorem 3.1. *Let $Y(t),Z(t) \in L_2^n(\bar{D})$. Under assumption 3.4, the following holds:*

$$\begin{aligned}&\Big\langle Y(t), F_X\big(t,X(t),\bar{U}_d(t)\big)Z(t)\Big\rangle - \Big\langle Z(t), F_X^*\big(t,X(t),\bar{U}_d(t)\big)Y(t)\Big\rangle \\ &= \Big\langle Y(t), N_X\big(t,X(t),\bar{U}_b(t)\big)Z(t)\Big\rangle_S - \Big\langle Z(t), N_X^*\big(t,X(t),\bar{U}_b(t)Y(t)\big)\Big\rangle_S\end{aligned} \tag{3.17}$$

The equivalent fields representation can be found in [68], and the proof is a standard result (c.f. [89]). The following corollary is a direct application of theorem 3.1 to the applicable terms of the HJB in eq. (3.16).

Corollary 3.2. *If assumption 3.4 holds, then*

$$\left\langle V_X, F_X^\top \delta X \right\rangle = \left\langle \delta X, F_X^* V_X \right\rangle - \left\langle V_X, \Delta N \right\rangle_S - \left\langle V_X, N_{U_b}^\top \delta U_b \right\rangle_S - \left\langle \delta X, N_X^* V_X \right\rangle_S \ , \quad (3.18)$$

and

$$\begin{aligned}\left\langle V_{XX}\delta X, F_X^\top \delta X' \right\rangle = \left\langle \delta X, F_X^* V_{XX}\delta X' \right\rangle - \left\langle V_{XX}\delta X, \Delta N \right\rangle_S - \left\langle V_{XX}\delta X, N_{U_b}^\top \delta U_b \right\rangle_S \\ - \left\langle \delta X, N_X^* V_{XX}\delta X' \right\rangle_S \ , \end{aligned} \quad (3.19)$$

where $\Delta N = N(X+\delta X, U_b+\delta U_b) - N(X,U_b)$, F_X^* is the adjoint operator of F_X, N_X^* is the adjoint operator of N_X, and we have suppressed explicit time dependencies for simplicity.

Plugging equations eqs. (3.18) and (3.19) into eq. (3.16) yields

$$\begin{aligned}
&-\frac{\mathrm{d}}{\mathrm{d}t}\left(V + \left\langle V_X, \delta X \right\rangle + \frac{1}{2}\left\langle \delta X, V_{XX}\delta X' \right\rangle\right) \\
&= \inf_{\delta U_d, \delta U_b}\Bigg[L + \left\langle L_X, \delta X \right\rangle + \left\langle L_{U_d}, \delta U_d \right\rangle + \left\langle L_{U_b}, \delta U_b \right\rangle_S + \frac{1}{2}\left\langle \delta X, L_{XX}\,\delta X' \right\rangle \\
&\qquad + \frac{1}{2}\left\langle \delta U_d, \left(L_{U_d X} + L_{XU_d}^\top\right)\delta X' \right\rangle + \frac{1}{2}\left\langle \delta U_b, \left(L_{U_b X} + L_{XU_b}^\top\right)\delta X' \right\rangle_S \\
&\qquad + \frac{1}{2}\left\langle \delta U_d, L_{U_d U_d}\,\delta U_d' \right\rangle + \frac{1}{2}\left\langle \delta U_b, L_{U_b U_b}\,\delta U_b' \right\rangle_S + \left\langle \delta X, F_X^* V_X \right\rangle - \left\langle V_X, \Delta N \right\rangle_S \\
&\qquad - \left\langle V_X, N_{U_b}^\top \delta U_b \right\rangle_S - \left\langle \delta X, N_X^* V_X \right\rangle_S + \left\langle V_X, F_{U_d}^\top \delta U_d \right\rangle + \left\langle \delta X, F_X^* V_{XX}\delta X' \right\rangle \\
&\qquad - \left\langle \delta X, V_{XX}\Delta N \right\rangle_S - \left\langle \delta X, V_{XX} N_{U_b}^\top \delta U_b \right\rangle_S - \left\langle \delta X, N_X^* V_{XX}\delta X' \right\rangle_S \\
&\qquad + \left\langle \delta X, V_{XX} F_{U_d}^\top \delta U_d \right\rangle\Bigg].
\end{aligned} \quad (3.20)$$

The form of the HJB equation in eq. (3.20) now properly incorporates boundary information of the value functional. As shown in the subsequent section, the resulting optimal update δU_b^* leverages the first and second derivative of the value functional, which is expected in the context of the established DDP method in finite dimensions.

We note that the form of the HJB eq. (3.20) is remarkably different than that of [83]. The fundamental differences arise due to a) improper application of Green's theorem, as discussed in [68], and b) terms that are a result of a fundamental difference of reasoning followed in their derivation. For example, the expansions in [83] are quite different than the ones computed here, and may reflect an evolution in the DDP approach over decades of research.

3.4 Optimal Distributed and Boundary Control Solutions

We find two singleton Newton solutions to the HJB eq. (3.20); one for the optimal distributed control update δU_d^*, and one for the optimal boundary control update δU_b^*.

Theorem 3.3. *Under the stated assumptions, the optimal distributed update δU_d^* and the optimal boundary update δU_b^* are given in Hilbert spaces by*

$$\delta U_d^* = -L_{U_d U_d}^{-1}\Big(F_{U_d}^\top V_X + L_{U_d}\Big) - \frac{1}{2}L_{U_d U_d}^{-1}\Big(L_{U_d X} + L_{X U_d}^\top + 2F_{U_d}^\top V_{XX}\Big)\delta X \tag{3.21}$$

$$\delta U_b^* = -L_{U_b U_b}^{-1}\Big(L_{U_b} - N_{U_b}^\top V_X\Big) - \frac{1}{2}L_{U_b U_b}^{-1}\Big(L_{U_b X} + L_{X U_b}^\top - 2N_{U_b}^\top V_{XX}\Big)\delta X \tag{3.22}$$

where we have defined the inverse operators $L_{U_d U_d}^{-1} \in \mathscr{L}\big(L_2^k(D), L_2^k(D)\big)$ and $L_{U_b U_b}^{-1} \in \mathscr{L}\big(L_2^l(S), L_2^l(S)\big)$ by their respective inverse kernels, given by

$$L_{U_d U_d}^{-1}(t)W(t) = \int_D \bar{L}_{U_d U_d}(t,x,y)W(t,y)\mathrm{d}y \tag{3.23}$$

$$L_{U_b U_b}^{-1}W(t) = \int_S \bar{L}_{U_b U_b}(t,\xi,\eta)W(t,\eta)\mathrm{d}S_\eta \tag{3.24}$$

with $\bar{L}_{U_d U_d}(t,x,y)$ and $\bar{L}_{U_b U_b}(t,\xi,\eta)$ denoting the kernel function of the operator $L_{U_d U_d}^{-1}(t)$

and $L^{-1}_{U_dU_d}(t)$ *(resp.), and satisfying the property of inverses for kernels*

$$\int_D L_{U_dU_d}(t,x,y)\bar{L}_{U_dU_d}(t,y,x')\mathrm{d}y = I\delta(x-x') \tag{3.25}$$

$$\int_S L_{U_bU_b}(t,\xi,\eta)\bar{L}_{U_dU_d}(t,\eta,\xi')\mathrm{d}S_\eta = I\delta(\xi-\xi') \tag{3.26}$$

Proof. The result can be found by applying a Newton step (e.g. taking the respective partial derivative and setting equal to zero) of the HJB eq. (3.20). □

The equivalent expressions in fields notation expose the spatial integration that takes place in these calculations, and are provided for completeness

$$\begin{aligned}\delta U_d = &-\int_D \bar{L}_{U_dU_d}(t,x,y)\Big(F^\top_{U_d}(t,y,y')V_X(t,y')+L_{U_d}(t,y)\Big)\mathrm{d}y\\ &-\frac{1}{2}\int_D\int_D \bar{L}_{U_dU_d}(t,x,y)\Big(L_{U_dX}(t,y,y')+L^\top_{XU_d}(t,y,y')\\ &\qquad +2F_{U_d}(t,y,y'')V_{XX}(t,y'',y')\Big)\delta X(t,y')\mathrm{d}y\,\mathrm{d}y' \qquad (3.27)\\ \delta U_b = &-\int_S \bar{L}_{U_bU_b}(t,x,\eta)\Big(L_{U_b}(t,\eta)-N^\top_{U_b}(t,\eta,\eta')V_X(t,\eta')\Big)\mathrm{d}S_\eta\\ &-\frac{1}{2}\int_S\int_D \bar{L}_{U_bU_b}(t,x,\eta)\Big(L_{U_bX}(t,\eta,y)+L_{XU_b}(t,\eta,y)\\ &\qquad -2N^\top_{U_b}(t,\eta,\eta')V_{XX}(t,\eta',y)\Big)\delta X(t,y)\mathrm{d}y\,\mathrm{d}S_\eta \qquad (3.28)\end{aligned}$$

3.5 The Backward Value Functional Equations

The value functional is a backward equation according to the HJB eq. (3.5), and is separated by order into zeroth, first, and second order derivative of the value functional. We present the fields representations of these backward equations without cross terms for simplicity. The more general forms of the backward equations with cross terms have been derived, but are lengthy and are omitted due to length considerations.

Theorem 3.4. *Under the above stated assumptions, and with optimal control in fields representation given by eqs.* (3.27) *and* (3.28)*, the zeroth-order backward value functional*

equation is given by

$$
\begin{aligned}
&-\frac{\mathrm{d}}{\mathrm{d}t}V(t,X(t,x)) \\
&= L-\int_S V_X(t,\xi)^\top \Delta N(t,\xi)\mathrm{d}S_\xi \\
&\quad -\frac{1}{2}\int_D\int_D \Big(L_{U_d}^\top(t,y)+V_X^\top(t,x)F_{U_d}(t,x,y)\Big)\bar{L}_{U_dU_d}(t,y,y')\Big(L_{U_d}(t,y') \\
&\qquad\qquad +F_{U_d}^\top(t,y',y'')V_X(t,y'')\Big)\mathrm{d}y\,\mathrm{d}y' \\
&\quad -\frac{1}{2}\int_S\int_S \Big(L_{U_b}^\top(t,\xi')-V_X^\top(t,\xi)N_{U_b}(t,\xi,\xi')\Big)\bar{L}_{U_bU_b}(t,\xi',\eta)\Big(L_{U_b}^\top(t,\eta) \\
&\qquad\qquad -N_{U_b}^\top(t,\eta,\eta')V_X(t,\eta')\Big)\mathrm{d}S_{\xi'}\,\mathrm{d}S_\eta
\end{aligned}
\tag{3.29}
$$

with terminal condition

$$
V\big(t_f,X(t_f,x)\big)=\phi\big(t_f,X(t_f,x)\big), \tag{3.30}
$$

the first-order backward value functional equation is given by

$$
\begin{aligned}
&-\frac{\mathrm{d}}{\mathrm{d}t}V_X\big(t,X(t,x)\big) \\
&= L_X(t,x)+F_X^*(t,x,y)V_X(t,y) \\
&\quad -\int_D\int_D V_{XX}(t,x,x')F_{U_d}(t,x',y)\bar{L}_{U_dU_d}(t,y,y')\Big(L_{U_d}(t,y')+F_{U_d}^\top(t,y',y'')V_X(t,y'')\Big)\mathrm{d}y\,\mathrm{d}y' \\
&\quad +\int_S\int_S V_{XX}(t,\xi,\xi')N_{U_b}(t,\xi',\eta)\bar{L}_{U_bU_b}(t,\eta,\eta')\Big(L_{U_b}(t,\eta')-N_{U_b}^\top(t,\eta',\eta'')V_X(t,\eta'')\Big)\mathrm{d}S_\eta\,\mathrm{d}S'_\eta
\end{aligned}
\tag{3.31}
$$

with boundary and terminal conditions

$$
0=N_X^*(t,\xi,\eta)V_X(t,\eta)-\int_S V_{XX}(t,\xi,\eta)\Delta N(t,\eta)\mathrm{d}S_\eta \tag{3.32}
$$

$$
V_X\big(t_f,X(t_f,x)\big)=\phi_X\big(t_f,X(t_f,x)\big), \tag{3.33}
$$

and the second-order backward value functional equation is given by

$$
\begin{aligned}
-\frac{\mathrm{d}}{\mathrm{d}t}V_{XX}(t,x,y) = {} & L_{XX}(t,x,y) + F_X^*(t,x,y')V_{XX}(t,y',y) + \left[F_X^*(t,x,y')V_{XX}(t,y',y)\right]^\top \\
& - \int_D \int_D V_{XX}(t,x,x')F_{U_d}(t,x',y')\bar{L}_{U_dU_d}(t,y',y'')F_{U_d}^\top(t,y'',z)V_{XX}(t,z,y)\mathrm{d}y'\mathrm{d}y'' \\
& - \int_S \int_S V_{XX}(t,x,\xi)N_{U_b}(t,\xi,\xi')\bar{L}_{U_bU_b}(t,\xi',\eta)N_{U_b}^\top(t,\eta,\eta')V_{XX}(t,\eta',y)\mathrm{d}S'_\xi\,\mathrm{d}S_\eta,
\end{aligned} \tag{3.34}
$$

with boundary and terminal conditions

$$
0 = N_X^*(t,\xi,\eta)V_{XX}(t,\eta,y) \tag{3.35}
$$

$$
V_{XX}\big(t_f, X(t_f,x)\big) = \phi_{XX}\big(t_f, X(t_f,x)\big) \tag{3.36}
$$

Proof. These equations are obtained by plugging in eqs. (3.21) and (3.22) into the eq. (3.20) and grouping terms by order of δX. □

The iterative forward-backward system is completed by the approximate variation of the field and boundary dynamics, which are found by rearranging eqs. (3.13) and (3.14) as

$$
\begin{aligned}
\frac{\mathrm{d}\delta X(t,x)}{\mathrm{d}t} &= F(\bar{X}+\delta X, \bar{U}_d+\delta U_d, t, x) - F(\bar{X}, \bar{U}_d, t, x) \\
&= F_X^\top(\bar{X}, \bar{U}_d, t, x)\delta X + F_{U_d}^\top(\bar{X}, \bar{U}_d, t, x)\delta U_d, \quad x \in D
\end{aligned} \tag{3.37}
$$

$$
\begin{aligned}
0 &= N(\bar{X}+\delta X, \bar{U}_b+\delta U_b, t, \xi) - N(\bar{X}, \bar{U}_b, t, \xi) \\
&= N_X^\top(\bar{X}, \bar{U}_b, t, \xi)\delta X + N_{U_b}^\top(\bar{X}, \bar{U}_b, t, \xi)\delta U_b, \quad \xi \in S.
\end{aligned} \tag{3.38}
$$

Finally, the control updates of iteration $k+1$ are given by

$$
U_d^{k+1} = U_d^k + \gamma_d \delta U_d^k \tag{3.39}
$$

$$
U_b^{k+1} = U_b^k + \gamma_b \delta U_b^k. \tag{3.40}
$$

3.6 Recovering Standard Results

The optimal distributed and boundary control and resulting backward value functional equations represent a generalization of a) DDP in finite dimensions and b) the LQR for PDEs. These results are standard results in the control literature, and as such it is important to the validity of our approach to clearly demonstrate that these standard results can be recovered from the equations detailed in the previous sections.

3.6.1 Differential Dynamic Programming in Finite Dimensions

We begin by roughly outlining an analogous derivation of DDP in finite dimensions. There are many different formulations of DDP in finite dimensions. Our approach specifically follows a body of literature that expands the pertinent functionals around a nominal trajectory. Despite having an extra term for a terminal constraint, we refer to [87] as they present a clean derivation that represents a finite dimensional analogue to the derivation in this document. We ignore the terms having to do with the terminal constraint and the terms that come from second order expansions of the dynamics for ease of comparison. Therein they consider a finite dimensional system of the general form

$$\frac{\mathrm{d}}{\mathrm{d}t}x = F(x,u,t), \quad x(t_0) = x_0 \tag{3.41}$$

The optimization problem is formulated as

$$\begin{aligned} V(x_0,t_0) &= \inf_u J(x,u) \\ &= \inf_u \left[\phi\left(x(t_f),t_f\right) + \int_{t_0}^{t_f} L(x,u,t)\mathrm{d}t \right] \end{aligned} \tag{3.42}$$

After applying standard Taylor expansions of the value functional, its first and second derivative, the dynamics, and the cost functional, plugging them into the HJB equation and

performing Newton minimization, they obtain the optimal control update as

$$\delta u = -L_{uu}^{-1}\left(L_u + F_u V_x\right) - \frac{1}{2}L_{uu}^{-1}\left(L_{ux} + L_{xu}^{\top} + 2F_u V_{XX}\right)\delta x \tag{3.43}$$

This is equivalent in form to the optimal distributed and boundary control update in Hilbert spaces given in eqs. (3.21) and (3.22). The resulting backward equations of the value functional in [87] are given by

$$-\frac{\mathrm{d}}{\mathrm{d}t}V = L - \frac{1}{2}k^{\top}L_{uu}k \tag{3.44}$$

$$-\frac{\mathrm{d}}{\mathrm{d}t}V_x = L_x + F_x V_x - K^{\top}L_{uu}k \tag{3.45}$$

$$-\frac{\mathrm{d}}{\mathrm{d}t}V_{XX} = L_{xx} - K^{\top}L_{uu}K + V_{xx}F_x^{\top} + F_x V_{xx} \tag{3.46}$$

where $k \in \mathbb{R}^k$, and $K \in \mathbb{R}^{k\times n}$ are given by

$$k = -L_{uu}^{-1}\left(L_u + F_u V_x\right) \tag{3.47}$$

$$K = -\frac{1}{2}L_{uu}^{-1}\left(L_{ux} + L_{xu}^{\top} + 2F_u V_{XX}\right) \tag{3.48}$$

In order to make the same comparison for the backward equations of the zeroth, first, and second-order value functional in fields, we first define the kernel functions $k_d : T \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^k$, $k_b : T \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^l$, $K_d : T \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \to T \times \mathbb{R}^k \times \mathbb{R}^n$, and $K_d : T \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \to T \times \mathbb{R}^l \times \mathbb{R}^n$, which are defined analogously to k and K in [87],

and given by

$$k_d(t,x,y) = -\bar{L}_{U_dU_d}(t,x,y)\Big(L_{U_d}(t,y) + F^{\top}_{U_d}(t,y,y')V_X(t,y')\Big) \tag{3.49}$$

$$k_b(t,x,\eta) = -\bar{L}_{U_bU_b}(t,x,\eta)\Big(L_{U_b}(t,\eta) - N^{\top}_{U_b}(t,\eta,\eta')V_X(t,\eta')\Big) \tag{3.50}$$

$$K_d(t,x,y,y') = -\frac{1}{2}\bar{L}_{U_dU_d}(t,x,y)\Big(L_{U_dX}(t,y,y') + L^{\top}_{XU_d}(t,y,y') + 2F_{U_d}(t,y,y'')V_{XX}(t,y',y'')\Big) \tag{3.51}$$

$$K_b(t,\xi,\eta,y) = -\frac{1}{2}\bar{L}_{U_bU_b}(t,x,\eta)\Big(L_{U_bX}(t,\eta,y) + L_{XU_b}(t,\eta,y) - 2N^{\top}_{U_b}(t,\eta,\eta')V_{XX}(t,\eta',y)\Big) \tag{3.52}$$

Thus eqs. (3.29), (3.31) and (3.34) in fields representation take the form

$$\begin{aligned} -\frac{\mathrm{d}}{\mathrm{d}t}V\big(t,X(t,x)\big) = L &- \int_S V_X(t,\xi)^{\top}\Delta N(t,\xi)\mathrm{d}S_\xi \\ &- \frac{1}{2}\int_D\int_D\int_D k_d^{\top}(t,x,y)L_{U_dU_d}(t,y,y'')k_d(t,y'',y')\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}y' \\ &- \frac{1}{2}\int_S\int_S\int_S k_b^{\top}(t,\xi,\xi')L_{U_bU_b}(t,\xi',\eta')k_b(t,\eta',\eta)\mathrm{d}S_\xi\,\mathrm{d}S'_\xi\,\mathrm{d}S_\eta \end{aligned} \tag{3.53}$$

$$\begin{aligned} -\frac{\mathrm{d}}{\mathrm{d}t}V_X\big(t,X(t,x)\big) = L_X + F_X^*V_X &\\ &- \int_D\int_D\int_D K_d^{\top}(t,x,y,y')L_{U_dU_d}(t,y',z)k_d(t,z,y'')\mathrm{d}y\,\mathrm{d}y'\,\mathrm{d}y'' \\ &- \int_S\int_S\int_S K_b^{\top}(t,x,\xi,\eta,)L_{U_bU_b}(t,\eta,\varphi)k_b(t,\varphi,\eta')\mathrm{d}S_\xi\,\mathrm{d}S_\eta\,\mathrm{d}S'_\eta \end{aligned} \tag{3.54}$$

$$\begin{aligned} -\frac{\mathrm{d}}{\mathrm{d}t}V_{XX}(t,x,y) = L_{XX}(t,x,y) + F_X^*(t,x,y')V_{XX}(t,y',y) + \big[F_X^*(t,x,y')V_{XX}(t,y',y)\big]^{\top} &\\ &- \int_D\int_D\int_D K_d(t,x,y,y')^{\top}L_{U_dU_d}(t,y',z)K_d(t,z,y'',y''')\mathrm{d}y'\,\mathrm{d}y''\,\mathrm{d}y''' \\ &- \int_S\int_S\int_S K_b(t,x,\xi,\eta)^{\top}L_{U_bU_b}(t,\eta,\varphi)K_b(t,\varphi,\eta',y)\mathrm{d}S_\xi\mathrm{d}S_\eta\mathrm{d}S'_\eta \end{aligned} \tag{3.55}$$

where in each equation, one of the integrals cancels due to the inverse kernel property in eqs. (3.25) and (3.26). Thus one can recover equations eqs. (3.44) to (3.46) by considering an ODE system that a) does not have a spatial state vector so the Volterra-Taylor expansion becomes a Taylor expansion and the volume integrals are equal to their integrand, b) does not have a spatial boundary so surface integrals over the boundary are zero, and c) has real-valued finite dimensional Jacobians defined on an orthonormal basis (with an orthonormal dual basis) so that the adjoint is equal to the transpose.

Thus, the DDP equations for PDEs are a generalization of the DDP equations for finite ODE systems. In the following section we demonstrate a similar generalization of the LQR solution for PDEs.

3.6.2 The Linear Quadratic Regulator of Fields

The linear quadratic regulator equations are obtained in [68]. Therein, they consider a linear PDE of the form

$$\partial_t X(t,x) = A_x(t)X(t,x) + B_d(t,x)U_d(t,x), \quad x \in D \tag{3.56}$$

$$X(t_0,x) = X_0 \tag{3.57}$$

where A_x is a linear differential operator that has standard form

$$A_x(t) = \sum_{i,j=1}^{n} A_{ij}(t,x)\frac{\partial^2}{\partial x_i \partial x_j} + \sum_{i=1}^{n} B_i(t,x)\frac{\partial}{\partial x_i} + C(t,x). \tag{3.58}$$

The boundary condition is given by

$$B_b(t,\xi)U_b(t,\xi) = F(t,\xi)X(t,\xi) + \sum_{j=1}^{n} A_j(t,\xi)\frac{\partial X}{\partial x_j}, \quad \xi \in S \tag{3.59}$$

where the operator A_j is given by

$$A_j(t,\xi) = \sum_{i=1} A_{ij}(t,\xi)\cos(n_\xi, x_i), \tag{3.60}$$

where (n_ξ, x_i) is the angle between the outward normal n_ξ at a boundary point $\xi \in S$ and the x_i-axis. The dynamics are equivalently described in Hilbert spaces as

$$\frac{\mathrm{d}}{\mathrm{d}t}X(t) = A(t)X(t) + B_d(t)U_d(t), \quad X \in L_2^n(D) \tag{3.61}$$

$$B_b(t)U_b(t) = F(t)X(t) + A(t)\cdot\nabla_x X(t), \quad X \in L_2^n(S) \tag{3.62}$$

which is a familiar control affine linear system form in Hilbert spaces. The optimization problem is formulated as

$$J(t_0, X_0, U_d, U_b) = \phi\big(X(t_f)\big) + \int_{t_0}^{t_f} L\big(t, X(t), U_d(t), U_b(t)\big)\mathrm{d}t \tag{3.63}$$

where the running cost L has the form

$$\begin{aligned} L\big(t, X(t), U_d(t), U_b(t)\big) = &\frac{1}{2}\int_D\int_D X(t,x)^\top Q(t,x,y)X(t,y)\mathrm{d}x\mathrm{d}y \\ &+ \frac{1}{2}\int_D\int_D U_d(t,x)^\top R_d(t,x,y)U_d(t,y)\mathrm{d}x\mathrm{d}y \\ &+ \frac{1}{2}\int_S\int_S U_b(t,\xi)^\top R_b(t,\xi,\eta)U_b(t,\eta)\mathrm{d}S_\xi\mathrm{d}S_\eta \end{aligned} \tag{3.64}$$

where the kernels $Q \geq 0$, $R_d > 0$ and $R_b > 0$ are all assumed to be symmetric about all spatial axes.

The resulting optimal distributed and boundary control equations are obtained after applying Green's theorem to the HJB equation and performing Newton minimization. They

Table 3.1: Corresponding Hilbert space operators between LQR of fields and DDP of fields.

DDP Operators	LQR Operators
$L_{U_dU_d}(t)$	$R_d(t)$
$L_{U_bU_b}(t)$	$R_b(t)$
$L_{XX}(t)$	$Q(t)$
$F_{U_d}(t)$	$B_d(t)$
$-N_{U_b}(t)$	$B_b(t)$
$V_X(t)$	$P(t)X(t)$
$V_{XX}(t)$	$P(t)$

are given by

$$U_d^*(t) = -R_d^{-1}(t)B_d(t)^\top P(t)X(t) \tag{3.65}$$

$$U_b^*(t) = -R_b^{-1}(t)B_b(t)^\top P(t)X(t) \tag{3.66}$$

which are equivalently written in fields representation as

$$U_d^*(t,x) = -\int_D \int_D \bar{R}_d(t,x,y)B_d^\top(t,y)P(t,y,y')X(t,y')\mathrm{d}y\mathrm{d}y' \tag{3.67}$$

$$U_b^*(t,x) = -\int_S \int_D \bar{R}_b(t,\xi,\eta)B_b^\top(t,\eta)P(t,\eta,y)X(t,y)\mathrm{d}y\mathrm{d}S_\eta \tag{3.68}$$

In order to make the generalization clear, we rewrite eqs. (3.65) and (3.66) in our notation using the correspondences listed in table 3.1

$$U_d^*(t) = -L_{U_dU_d}^{-1}(t)F_{U_d}^\top(t)V_X(t,X) \tag{3.69}$$

$$U_b^*(t) = L_{U_bU_b}^{-1}(t)N_{U_b}^\top(t)V_X(t,X) \tag{3.70}$$

and repeat eqs. (3.21) and (3.22) here for clarity

$$\delta U_d^* = -L_{U_dU_d}^{-1}\left(L_{U_d} + F_{U_d}^\top V_X\right) - \frac{1}{2}L_{U_dU_d}^{-1}\left(L_{U_dX} + L_{XU_d}^\top + 2F_{U_d}^\top V_{XX}\right)\delta X$$

$$\delta U_b^* = -L_{U_bU_b}^{-1}\left(L_{U_b} - N_{U_b}^\top V_X\right) - \frac{1}{2}L_{U_bU_b}^{-1}\left(L_{U_bX} + L_{XU_b}^\top - 2N_{U_b}^\top V_{XX}\right)\delta X$$

Thus, we can recover eqs. (3.69) and (3.70) by a) assuming the cost functional is a pure quadratic without cross terms so that the terms L_{U_d}, L_{U_b}, L_{XU_d}, L_{U_dX}, L_{XU_b}, and L_{U_bX} are null and b) using only gradient information of the value functional so that V_{XX} terms are ignored.

The resulting second-order backward value functional equation of Riccati type for LQR is given in fields representation by

$$\begin{aligned}\frac{\partial P(t,x,y)}{\partial t} &= -A_X^*(t)P(t,x,y) - \left[A_X^*(t)P(t,x,y)\right]^\top - Q(t,x,y) \\ &\quad + \int_D\int_D P(t,x,x')B_d(t,x')\bar{R}_d(t,x',x'')B_d^\top(t,x'')P(t,x'',y)\mathrm{d}x'\mathrm{d}x'' \\ &\quad + \int_S\int_S P(t,x,\xi)B_b(t,\xi)\bar{R}_b(t,\xi,\eta)B_b^\top(t,\eta)P(t,\eta,y)\mathrm{d}S_\xi\mathrm{d}S_\eta\end{aligned} \tag{3.71}$$

which is rewritten in the DDP notation by again applying the correspondences listed in table 3.1 as

$$\begin{aligned}\frac{\partial V_{XX}(t,x,y)}{\partial t} &= -F_X^*(t,x,y)V_{XX}(t,x,y) - \left[F_X^*(t,x,y)V_{XX}(t,x,y)\right]^\top - L_{XX}(t,x,y) \\ &\quad + \int_D\int_D V_{XX}(t,x,x')F_{U_d}(t,x')\bar{L}_{U_dU_d}(t,x',x'')F_{U_d}^\top(t,x'')V_{XX}(t,x'',y)\mathrm{d}x'\mathrm{d}x'' \\ &\quad + \int_S\int_S V_{XX}(t,x,\xi)F_{U_b}(t,\xi)\bar{L}_{U_bU_b}(t,\xi,\eta)F_{U_b}^\top(t,\eta)V_{XX}(t,\eta,y)\mathrm{d}S_\xi\mathrm{d}S_\eta.\end{aligned} \tag{3.72}$$

eq. (3.72) is identical to the second order backward value functional of DDP of fields without cross terms in the running cost, given in eq. (3.34).

Thus we conclude that the equations of STDDP are a generalization of LQR of fields. This generalization is analogous to the similar generalization of LQR of ODE systems to DDP of ODE systems. Whereas LQR is the analytically optimal controller for linear systems, it cannot be applied directly to a nonlinear system, nor can it be applied directly to a linear system whose running cost functional is not purely quadratic. In contrast, the iterative approximate optimal control method provided by DDP of fields was constructed for such systems.

3.7 Continuous-Time Convergence Analysis

Global convergence of the discrete finite dimensional DDP algorithm defined for discrete ODE systems was first provided by Yakowitz and Rutherford [86]. Later the proof that discrete finite dimensional DDP converges quadratically in the number of iterations was proved independently by Pantoja [90] and Murray and Yakowitz [91]. This quadratic convergence proof relied on convergence of Newton's method, but later an independent proof relying only on the dynamic programming principle was given by Liao and Shoemaker [85]. Through decades of application of the DDP algorithm, there have been numerous extensions of the proof of global convergence, for example for DDP on Lie groups in [92] and for DDP with generalized Polynomail Chaos expansions in [81]. However it appears to the best knowledge of the authors that most if not all proofs of global convergence are for DDP and its extensions in discrete time, and not in continuous time.

Typically, one determines provable convergence characteristics by investigating the behavior of the variation

$$\Delta J^i\big(t,X(t),U(t)\big) = \frac{\mathrm{d}J^i\big(t,X(t),U(t)\big)}{\mathrm{d}U^i_{t_0:t_f}}\Delta U^i_{t_0:t_f} \tag{3.73}$$

where $\Delta J^i\big(t,X(t),U(t)\big) := J^i\big(t,X(t),U(t)\big) - J^{i-1}\big(t,X(t),U(t)\big)$, and due to the decoupled nature of the distributed and boundary control updates, we have defined the Hilbert space control vector $U(t) \in L_2^{k+l}(\bar{D})$ as the direct product Hilbert space analog of the stacked distributed and boundary control vectors in fields representation $U(t,x) = [U_d(t,x), U_b(t,x)]^\top$. This notation simplifies our analysis significantly. We have also introduced the trajectory notation, where subscript $t_0 : t_f$ represents the entire trajectory in time of the associated variable, and used the superscript i for the STDDP iteration index. Similarly $\delta U^i_{t_0:t_f}$ denotes the control update trajectory for control trajectory $U^i_{t_0:t_f}$. This trajectory notation defines a *temporal* Hilbert space over time-indexed *spatial* Hilbert spaces. Let $L_2^n(T)$ denote the

Hilbert space of $L_2^n(\bar{D})$-vector functions square integrable over T with inner product

$$\left\langle X_{1,t_0:t_f}, X_{2,t_0:t_f} \right\rangle_T = \int_{t_0}^{t_f} \left\langle X_1(s), X_2(s) \right\rangle \mathrm{d}s. \tag{3.74}$$

This allows us to write time integrals over trajectory variables as inner product tensor contractions, and treat continuous trajectories as objects in a similar way to the continuum of the PDE variables. We begin by stating the following lemma, assumption, and proposition that will be used in our analysis.

Lemma 3.5. *Assume the cost functional has the form of eq.* (3.2) *and define the measurable backward recursive functional* $\psi \in \mathscr{L}\left(L_2^n(\bar{D})\right)$ *for some* $\varepsilon > 0$ *as*

$$\psi\big(t, X(t), U(t)\big) = \int_t^{t+\varepsilon} L_X\big(s, X(s), U(s)\big)\mathrm{d}s + \Phi(t,s)\psi\big(t+\varepsilon, X(t+\varepsilon), U(t+\varepsilon)\big) \tag{3.75}$$

$$\psi\big(t_f, X(t_f), U(t_f)\big) = \phi_X\big(t, X(t)\big) \tag{3.76}$$

where $\Phi(\cdot,\cdot)$ *is a contractive linear semigroup generated by the approximate variation dynamics in eqs.* (3.37) *and* (3.38)*, and is assumed to be positive definite almost everywhere. Then the cost functional satisfies*

$$\frac{\mathrm{d}J^i\big(t, X(t), U(t)\big)}{\mathrm{d}U^i_{t_0:t_f}} = \left\langle L_{U,t_0:t_f}, \mathbb{1}_{t_0:t_f} \right\rangle_T + \left\langle F_{U,t_0:t_f}, \psi_{t_0:t_f} \right\rangle_T \tag{3.77}$$

where $\mathbb{1}_{t_0:t_f}$ *is the trajectory of ones.*

Proof. We start with the total derivative for the cost functional

$$\frac{\mathrm{d}J^i\big(t,X(t),U(t)\big)}{\mathrm{d}U^i_{t_0:t_f}} = \frac{\partial J^i\big(t,X(t),U(t)\big)}{\partial U^i_{t_0:t_f}} + \frac{\partial J^i\big(t,X(t),U(t)\big)}{\partial X^i_{t_0:t_f}} \frac{\partial X^i_{t_0:t_f}}{\partial U^i_{t_0:t_f}} \tag{3.78}$$

$$= \Big\langle L_{U,t_0:t_f}, \mathbb{1}_{t_0:t_f} \Big\rangle_T + \Big\langle L_{X,t_0:t_f}, \frac{\partial X^i_{t_0:t_f}}{\partial U^i_{t_0:t_f}} \Big\rangle_T + \Big\langle \phi_X^\top\big(t_f, X(t_f)\big), \frac{\partial X^i_{t_f}}{\partial U^i_{t_0:t_f}} \Big\rangle \tag{3.79}$$

The state trajectory $X_{t_0:t_f}$ is due to the approximate state evolution given by eqs. (3.13) and (3.14), which has linear affine form with solution

$$X(t) = \Phi(t,t_0)X_0 + \int_{t_0}^{t} \Phi(t,s)F_U(s)U(s)\mathrm{d}s \tag{3.80}$$

Thus, one has

$$\frac{\mathrm{d}J^i\big(t,X(t),U(t)\big)}{\mathrm{d}U^i_{t_0:t_f}} = \Big\langle L_{U,t_0:t_f}, \mathbb{1}_{t_0:t_f} \Big\rangle_T + \Big\langle \Phi^\top(t,t_0:t_f)L_{X,t_0:t_f}, F_{U,t_0:t_f} \Big\rangle_T$$
$$+ \Big\langle \Phi^\top(T,t_0:t_f)\phi_X^\top\big(t_f,X(t_f)\big), F_{U,t_0:t_f} \Big\rangle_T \tag{3.81}$$

Now, due to the terminal condition on ψ given in eq. (3.76), one has

$$\frac{\mathrm{d}J^i\big(t,X(t),U(t)\big)}{\mathrm{d}U^i_{t_0:t_f}} = \Big\langle L_{U,t_0:t_f}, \mathbb{1}_{t_0:t_f} \Big\rangle_T + \Big\langle \Phi^\top(t,t_0:t_f)L_{X,t_0:t_f}, F_{U,t_0:t_f} \Big\rangle_T$$
$$+ \Big\langle \Phi^\top(T,t_0:t_f)\psi\big(t_f,X(t_f),U(t_f)\big), F_{U,t_0:t_f} \Big\rangle_T. \tag{3.82}$$

Finally, due to the backward recursion over the trajectory given by eq. (3.75), one has

$$\frac{\mathrm{d}J^i\big(t,X(t),U(t)\big)}{\mathrm{d}U^i_{t_0:t_f}} = \Big\langle L_{U,t_0:t_f}, \mathbb{1}_{t_0:t_f} \Big\rangle_T + \Big\langle \psi_{t_0:t_f}, F_{U,t_0:t_f} \Big\rangle_T, \tag{3.83}$$

which concludes the proof. □

Assumption 3.5. *The search space of control trajectories $\mathscr{U} \ni U^i_{t_0:t_f}$ is compact.*

Our analysis is simplified by the Q functional notation defined as follows

$$\begin{aligned} Q_{UU} &= L_{UU} & Q_U &= L_U + F_U^\top V_X \\ Q_{UX} &= L_{UX} + F_U^\top V_{XX} & Q_X &= L_X + F_X^\top V_X \\ Q_{XX} &= L_{XX} + F_X^* V_{XX} + \left[F_X^* V_{XX}\right]^\top \end{aligned}$$

Proposition 3.1. *Let the PDE $D(t) \in L_2^n(\bar{D})$ have dynamics*

$$\frac{\mathrm{d}}{\mathrm{d}t} D(t) = -F_X^\top D(t) + Q_{UX}^\top Q_{UU}^{-1} Q_U \tag{3.84}$$

$$D(T) = 0 \tag{3.85}$$

Then $D(t)$ has weak backwards solutions defined in the Hadamard sense, and given by

$$D(t) = \int_T^t \Phi^\top(t,\tau) Q_{UX}^\top(\tau) Q_{UU}^{-1} Q_U(\tau) \mathrm{d}\tau \tag{3.86}$$

Proof. The existence of weak solutions is given by the assumption that solutions to ψ and V_X exist. The rest of the proof is immediate given that the dynamics are of semilinear form and have a zero terminal condition. □

Theorem 3.6. *Consider the continuous-time optimal control problem in eq.* (3.4) *subject to the dynamics in eqs.* (2.6) *and* (2.7) *with cost functional J having no cross terms for simplicity. Let $\bar{U}_{t_0:t_f} \in L_2^{k+l}(T)$ be a nominal control trajectory and let $\delta U_{t_0:t_f} \in L_2^{k+l}(T)$ be the trajectory of control updates from eqs.* (3.21) *and* (3.22)*. Then the following holds:*

$$\Delta J^i\big(t, X(t), U(t)\big) = -\gamma \Big\langle Q_{U,t_0:t_f}, M_{t_0:t_f} Q_{U,t_0:t_f} \Big\rangle_T + O(\gamma^2) \tag{3.87}$$

where the trajectory operator $M_{t_0:t_f} \in \mathscr{L}\big(L_2^{l+k}(T), L_2^{l+k}(T)\big)$ has a positive definite kernel $\forall t \in T$.

Proof. Observe that due to the iterative updates in eqs. (3.39) and (3.40), we have

$$\Delta U^i_{t_0:t_f} = \gamma \delta U^i_{t_0:t_f}. \tag{3.88}$$

Also at our disposal is the identity

$$L_{U,t_0:t_f} + F^\top_{U,t_0:t_f} \psi_{t_0:t_f} = Q_{U,t_0:t_f} - F^\top_{U,t_0:t_f} \big(V_{X,t_0:t_f} - \psi_{t_0:t_f}\big). \tag{3.89}$$

Plugging eq. (3.89) into eq. (3.77) yields

$$\begin{aligned}
\Delta J^i\big(t, X(t), U(t)\big) = &-\gamma \Big\langle Q_{U,t_0:t_f}, Q^{-1}_{UU,t_0:t_f} Q_{U,t_0:t_f} \Big\rangle_T \\
&+ \gamma \Big\langle F^\top_{U,t_0:t_f} \big(V_{X,t_0:t_f} - \psi_{t_0:t_f}\big), Q^{-1}_{UU,t_0:t_f} Q_{U,t_0:t_f} \Big\rangle_T \\
&- \gamma \Big\langle Q_{U,t_0:t_f}, Q^{-1}_{UU,t_0:t_f} Q_{UX,t_0:t_f} \delta X_{t_0:t_f} \Big\rangle_T \\
&+ \gamma \Big\langle F^\top_{U,t_0:t_f} \big(V_{X,t_0:t_f} - \psi_{t_0:t_f}\big), Q^{-1}_{UU,t_0:t_f} Q_{UX,t_0:t_f} \delta X_{t_0:t_f} \Big\rangle_T
\end{aligned} \tag{3.90}$$

Note that the total variation $\delta X \in L^n_2(\bar{D})$, with dynamics given in semilinear form by eqs. (3.37) and (3.38), has a solution given by

$$\delta X(t) = \Phi(t, t_0) \delta X_0 + \int_0^t \Phi(t, s) F_U(s,) \delta U(s) \mathrm{d}s \tag{3.91}$$

Thus, since $\delta X_0 = 0$, and $\delta U(t) = O(\gamma)$, it follows that $\delta X = O(\gamma)$, so we have

$$\begin{aligned}
\Delta J^i\big(t, X(t), U(t)\big) = &-\gamma \Big\langle Q_{U,t_0:t_f}, Q^{-1}_{UU,t_0:t_f} Q_{U,t_0:t_f} \Big\rangle_T \\
&+ \gamma \Big\langle F^\top_{U,t_0:t_f} \big(V_{X,t_0:t_f} - \psi_{t_0:t_f}\big), Q^{-1}_{UU,t_0:t_f} Q_{U,t_0:t_f} \Big\rangle_T + O(\gamma^2)
\end{aligned} \tag{3.92}$$

Next, notice that $D(t) = V_X(t) - \psi(t)$ has dynamics of the form of eq. (3.84), with an equivalent terminal condition. Thus, plugging in eq. (3.86) into eq. (3.92) and reducing

yields

$$\Delta J^i\big(t,X(t),U(t)\big) = -\gamma\Big\langle Q_{U,t_0:t_f}, Q^{-1}_{UU,t_0:t_f} Q_{U,t_0:t_f}\Big\rangle_T - \gamma\Big\langle Q_{U,t_0:t_f}, A_{1,t_0:t_f} Q_{U,t_0:t_f}\Big\rangle_T + O(\gamma^2) \tag{3.93}$$

where $A_{1,t_0:t_f} \in \mathscr{L}\big(L_2^{k+l}(T), L_2^{k+l}(T)\big)$ has a positive definite kernel $\forall\, t \in T$ due to the positive definiteness of $\Phi(\cdot,\cdot)$ by definition, and the positive definiteness of V_{XX} by assumption. Thus, due to the positive definiteness of the kernels of $L_{U_dU_d}(t)$ and $L_{U_bU_b}(t)$ by assumption, one can form $M \in \mathscr{L}\big(L_2^{k+l}(\bar{D}), L_2^{k+l}(\bar{D})\big)$ with positive definite kernel $\forall\, t \in T$ such that

$$\Delta J^i\big(t,X(t),U(t)\big) = -\gamma\Big\langle Q_{U,t_0:t_f}, M_{t_0:t_f} Q_{U,t_0:t_f}\Big\rangle_T + O(\gamma^2) \tag{3.94}$$

which concludes the proof □

Corollary 3.7. *Suppose assumption 3.5 holds. Then the iterative eqs.* (2.6)*,* (2.7)*,* (3.21)*,* (3.22)*,* (3.29) *to* (3.36)*,* (3.39) *and* (3.40) *will converge to a stationary solution of eq.* (3.4)*.*

Proof. For simplicity of notation, we will use the shorthand $J^i := J^i\big(t,X^i(t),U^i(t)\big)$. Theorem 3.6 and assumption 3.5 together imply that $\exists \gamma \in [0,1)$ such that the change in the cost over iterations $\Delta J^i := J^i - J^{i-1} < 0$, $\forall i \in \mathbb{N}_+$. The cost functional $J(\cdot,\cdot,\cdot)$ is continuous in its arguments since it is differentiable by assumption and $\lim_{i\to\infty} \Delta J^i = 0$. Thus $\exists$ a pair $(X^*_{t_0:t_f}, U^*_{t_0:t_f})$ such that $\lim_{i\to\infty} J\big(t,X^i(t),U^i(t)\big) = J\big(t,X^*(t),U^*(t)\big) =: J^*$.

Next, $\lim_{i\to\infty} \Delta J^i = 0$ together with eq. (3.87) and the positive definiteness of $M_{t_0:t_f}$ imply that $\lim_{i\to\infty} Q^i_{U,t_0:t_f} = 0_{t_0:t_f}$. By this notation we mean that $\lim_{i\to\infty} Q^i_U\big(t,X(t),U(t)\big) = 0$ $\forall t \in T$. Recall also that δX^i_0. We seek the intermediate result that $\delta X^*_{t_0:t_f} := \lim_{i\to\infty} X^i_{t_0:t_f} = 0_{t_0:t_f}$. This can be observed in the coupled solutions of $\delta X(t)$ and $\delta U(t)$, but is more clear

by inspection of the closed-loop variation dynamics, which have generalized form

$$\frac{\mathrm{d}\delta X(t)}{\mathrm{d}t} = F_X^\top(t)\delta X(t) - F_U^\top(t)Q_{UU}^{-1}(t)\Big(Q_U(t) + Q_{UX}(t)\delta X(t)\Big) \tag{3.95}$$

$$= \Big(F_X^\top(t) - F_U^\top(t)Q_{UU}^{-1}(t)Q_{UX}(t)\Big)\delta X(t) - F_U^\top(t)Q_{UU}^{-1}(t)Q_U(t) \tag{3.96}$$

with solution of the form

$$\delta X(t) = \tilde{\Phi}(t,t_0)\delta X_0 - \int_{t_0}^{t} \tilde{\Phi}(t,s)F_U^\top(s)Q_{UU}^{-1}(s)Q_U(s)\mathrm{d}s, \tag{3.97}$$

where $\tilde{\Phi}(\cdot,\cdot)$ is a contractive linear semigroup. Thus, since $\delta X_0 = 0$, and $\lim_{i\to\infty} Q^i_{U,t_0:t_f} = 0$, we have that $\delta X^*_{t_0:t_f} = 0_{t_0:t_f}$ and thus $\delta U^*_{t_0:t_f} := \lim_{i\to\infty} \delta U^i_{t_0:t_f} = 0_{t_0:t_f}$, which implies that $\lim_{i\to\infty} U^i_{t_0:t_f} = U^*_{t_0:t_f}$.

Finally, we must show that the converged control trajectory $U^*_{t_0:t_f}$ is stationary. To show this, consider the rate of change of the cost functional with respect to control over iterations in the limit, namely

$$\lim_{i\to\infty} \frac{\mathrm{d}J^i}{\mathrm{d}U^i_{t_0:t_f}} = \lim_{i\to\infty} \Big\langle Q^i_{U,t_0:t_f}, \mathbb{1}_{t_0:t_f} \Big\rangle_T - \lim_{i\to\infty} \Big\langle F^i_{U,t_0:t_f}, V^i_{X,t_0:t_f} - \psi_{t_0:t_f} \Big\rangle_T \tag{3.98}$$

Since we already showed that $\lim_{i\to\infty} Q^i_{U,t_0:t_f} = 0$, one can easily apply the dominated convergence theorem to show that the first term is a zero trajectory. We must only prove that the second term is also a zero trajectory. By proposition 3.1, we have

$$\begin{aligned}
\lim_{i\to\infty} \Big(V^i_X(t) - \psi(t)\Big) &= \lim_{i\to\infty} \int_T^t \Phi^\top(t,\tau)Q^{i^\top}_{UX}(\tau)Q^{i^{-1}}_{UU}Q^i_U(\tau)\mathrm{d}\tau \\
&= \int_T^t \lim_{i\to\infty} \Phi^\top(t,\tau)Q^{i^\top}_{UX}(\tau)Q^{i^{-1}}_{UU}Q^i_U(\tau)\mathrm{d}\tau \\
&= 0
\end{aligned}$$

where we have again applied a properly formulated dominated convergence argument due to boundedness of $V_{XX,t_0:t_f}$ and $X_{t_0:t_f}$ $\forall i \in \mathbb{N}_+$ by assumption, and due to $Q_{U,t_0:t_f}$ being a

decreasing function over iterations. The limit is again zero since $\lim_{i\to\infty} Q^i_{U,t_0:t_f} = 0_{t_0:t_f}$. Thus $\lim_{i\to\infty} \frac{\mathrm{d}J^i}{\mathrm{d}U^i_{t_0:t_f}} = 0_{t_0:t_f}$, and the converged trajectory $U^*_{t_0:t_f}$ is indeed stationary, which concludes the proof. □

3.8 STDDP Algorithm

The resulting STDDP algorithm can be applied for control of any nonlinear forward spatio-temporal PDE system satisfying the stated assumptions. It is an iterative forward-backward approach, wherein each iteration forward propagates the dynamics, backward propagates the value functional and its derivatives, and updates the control based on approximate variation dynamics. The resulting procedure is described in greater detail in algorithm 1.

Note that algorithm 1 has a forward process, a backward process, and another forward process. While this is a simpler algorithmic exposition, the runtime performance can be improved by simply combining the two forward time loops. While the numerical experiments in this chapter were performed with a fixed learning rate for demonstration purposes, it can be numerically advantageous to apply line search methods to adapt the learn rate. Some such methods are described in [93] and [75], and typically evaluate the best learning rate based on the best improvement in the cost functional. However, since the value functional typically encodes problem information beyond the cost metric, one may also evaluate learning rate based on improvements in the value functional.

The inputs of the STDDP algorithm can change depending on the specific problem but in most cases contain time interval (T), number of iterations (K), initial state (X_0), time discretization (Δt), distributed control learn rate (γ_d), and boundary control learn rate (γ_b). One may also include a number of rollouts (R) for a parallelized line search. Instead of a fixed number of iterations, one may also check for convergence using relative or absolute convergence criteria in either the cost functional or the value functional [75].

Algorithm 1 STDDP

Function: $(U_d^*, U_b^*) = \boldsymbol{STDDP}(T, K, X_0, \bar{U}_d, \bar{U}_b, \Delta t, \gamma_d, \gamma_b)$
for $k = 1$ to K **do**
    Forward propagate PDE dynamics in eq. (2.6)
    Evaluate running cost L and its partial derivatives
    Evaluate terminal cost ϕ and its partial derivatives
    Backward propagate value functional via eqs. (3.29), (3.31) and (3.34)
    Forward propagate approximate variation dynamics via eqs. (3.37) and (3.38)
    Compute updates δU_d^k and δU_b^k via eqs. (3.21) and (3.22)
    Update control U_d^{k+1}, U_b^{k+1} via eqs. (3.39) and (3.40)
end for

3.8.1 Forward & Backward PDE Discretization Methods

In order to implement the forward spatio-temporal system dynamics in eqs. (2.6) and (2.7) and the backward value functional system in eqs. (3.29) to (3.36) on a digital computer, these forward and backward PDEs must be spatially and temporally discretized.

Nonlinear PDEs in the Eulerian formalism are often spatially discretized using either finite difference methods, Galerkin methods, or finite element methods. In this work we apply a spatial central finite difference discretization, which yields a fixed 1D grid of length a, with J elements. We note that through the above derivation, any discretization can be used in place of the central difference.

While there are numerous works describing temporal discretization methods for a multitude of forward PDEs, there are relatively few that describe temporal discretization schemes for backward PDEs of Riccati type. In finite dimensions, these are typically referred to as Riccati Differential Equations (RDEs), and their discretization presents several difficulties which stem from a matrix-valued variable that cannot be analytically isolated without using a Kronecker scheme. Furthermore, RDEs are known to be quite stiff in many contexts due to a fast transient response [94].

The most straightforward method is the explicit time Euler discretization method, which has a fast implementation, yet is sensitive to discretization step size for stiff dynamics. This sensitivity can be reduced by applying Runge-Kutta time-integration techniques, however

one must either super-sample the dynamics or apply an equivalent Runge-Kutta integration for the dynamics and value functionals.

Semi-implicit time discretization and implicit time discretization are well known to handle stiff dynamics, yet require isolation of the value functional. This in turn yields an update with a very large Kronecker sum matrix inversion. To elucidate, consider the discretized 1D Hilbert space representation of eq. (3.34), where $F_X^* = F_X^\top$, given by

$$\begin{aligned} -\frac{\mathrm{d}}{\mathrm{d}t}V_{XX}(t) = L_{XX} + \frac{1}{\Delta x}F_X^\top V_{XX}(t) + \frac{1}{\Delta x}V_{XX}(t)F_X - \frac{1}{\Delta x^2}V_{XX}(t)F_{U_d}L_{U_dU_d}^{-1}F_{U_d}^\top V_{XX}(t) \\ - \frac{1}{\Delta x^2}V_{XX}(t)N_{U_b}L_{U_bU_b}^{-1}N_{U_b}^\top V_{XX}(t). \end{aligned} \tag{3.99}$$

Clearly, the desired variable V_{XX} cannot be completely isolated in this form. However, one can equivalently write a vector form by application of the vec operator

$$\begin{aligned} -\mathrm{vec}\left(\frac{\mathrm{d}}{\mathrm{d}t}V_{XX}(t)\right) = \mathrm{vec}(L_{XX}) + \frac{1}{\Delta x}F_X^\top \oplus F_X^\top \mathrm{vec}\big(V_{XX}(t)\big) \\ - \frac{1}{\Delta x^2}\mathrm{vec}\big(V_{XX}(t)F_{U_d}L_{U_dU_d}^{-1}F_{U_d}^\top V_{XX}(t)\big) \\ - \frac{1}{\Delta x^2}\mathrm{vec}\big(V_{XX}(t)N_{U_b}L_{U_bU_b}^{-1}N_{U_b}^\top V_{XX}(t)\big). \end{aligned} \tag{3.100}$$

Semi-implicit time discretization schemes typically evaluate terms that are linear in $V_{XX}(t)$ at the current time step and non-linear terms in $V_{XX}(t)$ at the next time step [95], which is the previous time step in the case of backward PDEs. The resulting semi-implicit update is given by

$$\begin{aligned} &\mathrm{vec}\Big(V_{XX}(t_{k-1})\Big) \\ &= \Big[I - F_X^\top \otimes F_X^\top \Delta t\Big]^{-1}\Big[\mathrm{vec}\Big(V_{XX}(t_k)\Big) + \mathrm{vec}\Big(L_{XX}\Big)\Delta t - \mathrm{vec}\Big(V_{XX}(t_k)F_{U_d}L_{U_dU_d}^{-1}F_{U_d}^\top V_{XX}(t_k)\Big)\Delta t\Big]. \end{aligned} \tag{3.101}$$

The resulting update equation is less sensitive to time discretization step size Δt, however it requires the inversion of a large matrix of size $J^2 \times J^2$ for each time step of each iteration,

where J is the spatial discretization size of the 1D PDE. A key observation is that the matrix $M := I - F_X^\top \otimes F_X^\top \Delta t$ typically only has as many diagonals as the order of the spatial discretization, and is zeros elsewhere except for the boundary conditions, thus it is a sparse matrix. For example, in the case of a second order spatial central difference discretization of the Burgers equation with Homogeneous Dirichlet boundary conditions, M is tridiagonal. Thus the inverse can be efficiently computed with sparse linear equation solvers such as SuperLU [96].

In [94], the authors describe so called D-methods, which reduce computational complexity inherent to semi-implicit methods by applying explicit Euler discretization to some subset of the variables, and applies semi-implicit discretization to the rest. This could dramatically reduce complexity; if $J_e \leq J$ is the number of points treated with explicit discretization, then the resulting semi-implicit inverse is of size $(J - J_e)^2 \times (J - J_e)^2$. This may have dramatic benefit for ODEs systems where one may have slower and faster channels, However it is not clear how to select grid elements for the associated D-method for Riccati PDEs.

3.9 Simulated Experiments

We applied the STDDP algorithm to two simulated PDE experiments to optimally control the system to a prescribed desired behavior. Each experiment used less than 32 GB RAM, and was run on a desktop computer with an Intel Xeon 12-core CPU with a NVIDIA GeForce GTX 980 GPU. The computations did not utilize GPU parallelization, however many operations, such as cost and partial derivative computations, can be parallelized for greater computational efficiency.

The simulated experiments involve reaching tasks, where the PDE is initialized at a zero initial condition over the spatial region, and must reach certain field values at prescribed regions of the spatial domain. As discussed in the previous section, each PDE was spatially discretized by a spatial central difference discretization, and an expicit-time Euler discretization. The first and second derivative of the value functional were spatially

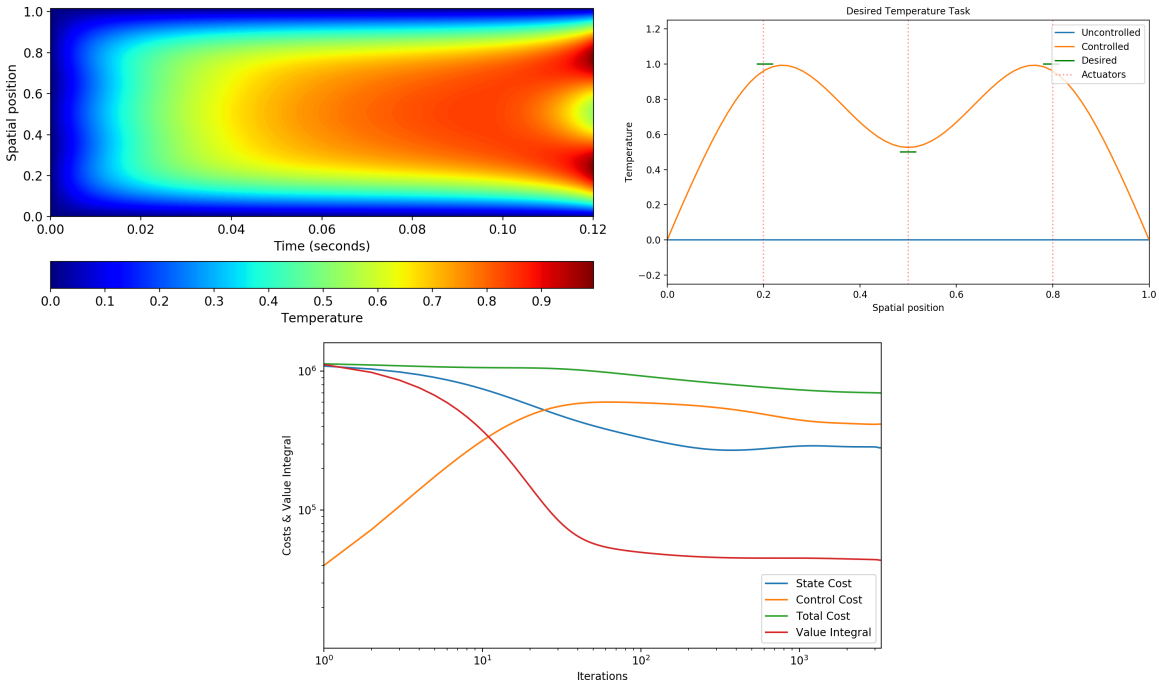


Figure 3.1: Heat Equation Temperature Reaching Task. (left) controlled contour plot where color represents temperature, (right) final time snapshot of the uncontrolled and controlled systems, (bottom) convergence plot of the heat equation temperature reaching task on a log-log scale, where the value integral depicted in red is the time integral of the value functional.

discretized on the same spatial central difference grid as the forward dynamical PDE, and all three backward equations were temporally discretized with an explicit Euler discretization. Regularization was added to the second derivative of the value functional in order to aid in numerical stability.

Each experiment considered a pure quadratic cost functional of the form

$$\begin{aligned} J\big(t,h(t,x),U_d(t),U_b(t)\big) := &\Big\langle h(t_f,x)-h_{\text{des}}(t_f,x), Q_f\big(h(t_f,x)-h_{\text{des}}(t_f,x)\big)\Big\rangle_{D_{\text{des}}} \\ &+\int_{t_0}^{t_f}\Big(\Big\langle h(t,x)-h_{\text{des}}(t,x), Q\big(h(t,x)-h_{\text{des}}(t,x)\big)\Big\rangle_{D_{\text{des}}} \\ &+\Big\langle U_d(t,x), R_d U_d(t,x)\Big\rangle + \Big\langle U_b(t,x), R_b U_b(t,x)\Big\rangle_S\Big)\,\mathrm{d}t, \end{aligned} \tag{3.102}$$

where the inner product $\langle\cdot,\cdot\rangle_{D_{\text{des}}}$ is defined on the desired subregion $D_{\text{des}} \subseteq D$.

The first experiment was a temperature reaching task on the 1D Heat equation with homogeneous Dirichlet boundary conditions, given in fields representation by

$$\begin{aligned} \partial_t h(t,x) &= \varepsilon \partial_{xx} h(t,x) + \mathbf{m}(\mathbf{x})^\top U_d(t,x), \\ h(t,0) &= h(t,a) = 0, \\ h(0,x) &= h_0(x), \end{aligned} \tag{3.103}$$

where ε is the thermal diffusivity parameter. The heat equation is a pure diffusion equation, and validates the approach's ability to achieve high quality distributed control solutions in the linear PDEs regime. The STDDP algorithm was run until convergence, and the results of which are depicted in fig. 3.1. Starting from a zero initial condition, the PDE was tasked with raising the temperature to $T = 1.0$ at the outer regions, and raising the temperature to $T = 0.5$ at the central region.

The system was temporally discretized into 1200 time steps and spatially discretized into 64 grid points. The typical convergence behavior for the STDDP algorithm applied to the heat equation is depicted in the bottom subfigure of fig. 3.1. In this case, the weight values were $R_d = 0.4$, $Q = 300$, and $Q_f = 300$. Depicted is a log-log plot of the cost functional $J\big(t, X(t)\big)$, its state cost functional and control cost functional components, and the time integral of the value functional, which is concisely termed the value integral. The convergence behavior of the value integral demonstrates super-quadratic convergence in the first 50 iterations.

The second experiment was a velocity reaching task on the 1D Burgers equation with non-homogenous Dirichlet boundary conditions, given in *fields representation* by

$$\begin{aligned} \partial_t h(t,x) &= -h(t,x)\partial_x h(t,x) + \varepsilon \partial_{xx} h(t,x) + \mathbf{m}(\mathbf{x})^\top U_d(t,x), \\ h(t,0) &= h(t,a) = 1.0, \\ h(0,x) &= h_0(x), \end{aligned} \tag{3.104}$$

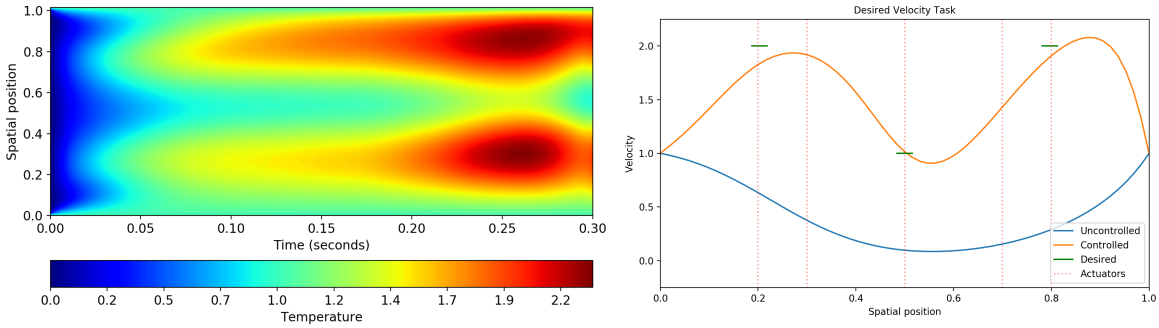


Figure 3.2: Burgers Equation Velocity Reaching Task. (left) controlled contour plot where color represents velocity, (right) final time snapshot comparing to the uncontrolled system.

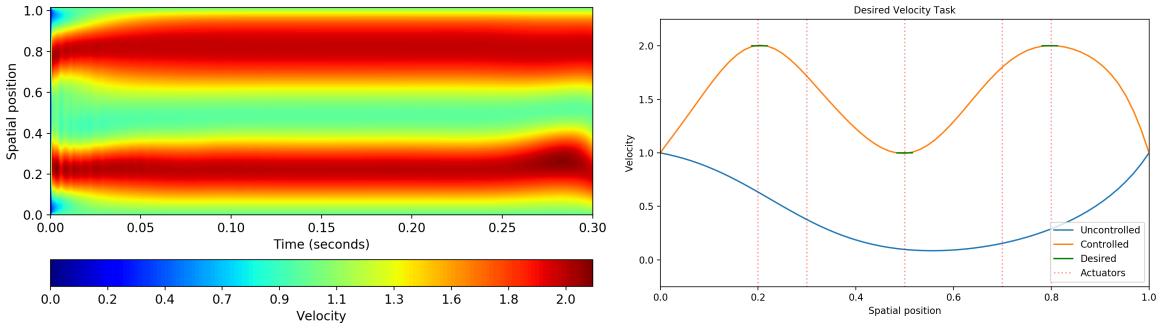


Figure 3.3: Burgers Equation Velocity Reaching Task with Simulated Annealing. (left) controlled contour plot where color represents velocity, (right) final time snapshot comparing the optimized solution to the uncontrolled system.

where the parameter ε is the viscosity of the medium. The Burgers equation is a nonlinear PDE, and demonstrates the efficacy of the approach on nonlinear PDEs. Starting from a zero initial condition, the PDE is tasked with raising the velocity to $v = 2.0$ on the outer regions, and $v = 1.0$ on the central region. The Burgers equation is often used as a simplified model of fluid flow, however also has applications in describing the dynamics of swarms for robotic systems [10]. The STDDP was applied to the Burgers PDE and was run until convergence. The results are depicted in fig. 3.2.

The nonlinear advection present in the Burgers equation produces an apparent rightward motion that builds over the spatial domain to create an apparent wavefront towards the right endpoint. The system is provided with 5 actuators, and must overcome this nonlinearity in order to minimize the state cost. Despite the added actuators, the task remains severely

under-actuated. In this case, the weight values were $R_d = 0.4$, $Q = 30$, and $Q_f = 30$. As depicted, the provided values of state and control cost weighting provide a balancing between the state and control performance metrics.

In both of the experiments, the various discretization schemes described in section 3.8.1 were tested, namely the explicit Euler discretization, a Runge-Kutta 2-point discretization, and the semi-implicit. The authors report that while the semi-implicit method had slightly lower sensitivity to the time-step increment Δt compared to the explicit Euler and Runge-Kutta methods, the large matrix inversion caused dramatically slower per-iteration run-time. The Runge-Kutta method had higher accuracy than the Euler method, but required super-sampling (i.e. sampling the midpoint of a time-increment) thus doubling the total time steps on forward and backward passes. The explicit Euler discretization had the fastest per-iteration run time at about 0.4 seconds per iteration, and was stabilized using regularization methods, akin to [93].

Common to finite and infinite dimensional DDP methods are parameter sensitivities that may limit choice of the control cost weighting and the state cost weightings. When these limits arise, they are typically due to the numerically stiff and sensitive dynamics found in the backwards Ricatti equation eq. (3.34), and present a limitation in the ability of DDP approaches to use arbitrary ratios of state performance and control effort. This can be especially limiting in systems with under-actuation as control signals can often be much larger for task completion, thus requiring a larger ratio between state cost weight and control cost weight. Without a "warm start", the operational initialization window for control weights may limit the use of an arbitrary desired set of parameters, thus changing the task specifications to meet numerical requirements.

In fig. 3.3, we demonstrate that this can be overcome with a simple simulated annealing scheme. In this simulated experiment, the simulated annealing scheme was adopted in order to reach an arbitrarily large weight ratio $W_d := Q/R_d = 4.8 \times 10^6$ starting from a nominal weight ratio of $W_d = 25$. This approach allows one to arbitrarily choose the

relative importance of state performance and control effort. Depicted is a contour plot that demonstrates that the desired regions are quickly reached, and the system remains at the desired region for the duration of the simulation. Also depicted is a final time snapshot with dramatically smaller deviation from the desired region as compared to the solution in fig. 3.2, albeit at the expense of larger control effort.

3.10 Discussion & Conclusion

We address the optimal control on nonlinear spatio-temporal systems through the lens of the Bellman principle of optimality, and develop the STDDP framework. We demonstrate that the resulting forward-backward system of equations can recover standard results, including the LQR solution for linear PDEs and the DDP solution for finite nonlinear ODEs. We analyze the convergence behavior and emerge with provable global convergence of the resulting forward-backward system. We discuss and develop discretization schemes for the backward second derivative of the value functional, and implement the resulting algorithm on a linear PDE system and a nonlinear PDE system.

The numerical results demonstrate the utility of the STDDP framework. It has the capability of obtaining high quality control solutions in the linear and nonlinear regime for spatio-temporal PDE systems. It has flexibility with respect to discretization schemes due to the optimize-then-discretize approach. It exhibits computationally efficiency for 1D PDEs with a typical 0.5 second time-per-iteration without any parallelization.

Overall, the results presented in this chapter are encouraging to the authors for future work on extending the approach to 2D and 3D problem spaces. Such scaling will result in large tensors, however one can leverage the sparsity inherent in PDE discretizations and utilize common tensor decompositions such as the tensor train decomposition [97] for a dramatic computational speed-up. Other future directions include extensions to the case of a system with additive Gaussian noise, second order expansions of the dynamics, and novel methods to handle the sensitivities that arise in the discretization of the backward process.

CHAPTER 4

LEVERAGING STOCHASTICITY FOR OPEN LOOP AND MODEL PREDICTIVE CONTROL OF SPATIO-TEMPORAL SYSTEMS

This chapter presents an open loop and MPC methodology for control of SPDEs related to fluid dynamics. Such systems are grounded on the theory of stochastic calculus in function spaces. The resulting architecture is not restricted to any particular finite representation of the original system; the control updates are independent of the method used to numerically simulate the SPDEs, which allows the most suitable problem dependent numerical scheme (e.g., finite differences, Galerkin methods, finite elements, etc.) to be employed.

Furthermore, deriving the variational optimization approach for optimal control entirely in Hilbert spaces overcomes numerical issues, including matrix singularities and SPDE space-time noise degeneracies that typically arise in finite dimensional representations of SPDEs. Thus, the work in this chapter is a generalization of ITC methods in finite dimensions [98, 55, 99, 100] to infinite dimensions and inherits crucial characteristics from its finite dimensional counterparts.

However, the primary benefit of the ITC approach presented in this chapter is that the stochasticity inherent in the system can be *leveraged* for control. Namely, The inherent system stochasticity is utilized for exploration in the space of trajectories of SPDEs in Hilbert spaces, which provide a Newton-type parameter update on the parametrized control policy. Importance sampling techniques are incorporated to iteratively guide the sampling distribution, and result in a mathematically consistent and numerically realizable sampling-based algorithm for distributed and boundary control of semi-linear SPDEs.

4.1 Problem Formulation

At the core of our method are comparisons between sampled stochastic paths used to perform Newton-type control updates, as depicted in fig. 4.1. Let, H, U be separable Hilbert spaces with inner products $\langle\cdot,\cdot\rangle_H$ and $\langle\cdot,\cdot\rangle_U$ resp,, σ-fields $\mathscr{B}(H)$ and $\mathscr{B}(U)$ resp. and probability space $(\Omega,\mathscr{F},\mathbb{P})$ with filtration $\mathscr{F}_t$, $t\in[0,T]$. Consider the controlled and uncontrolled infinite-dimensional stochastic systems of the form

$$\mathrm{d}X = \mathscr{A}X\mathrm{d}t + F(t,X)\mathrm{d}t + \frac{1}{\sqrt{\rho}}G(t,X)\mathrm{d}W(t), \tag{4.1}$$

$$\mathrm{d}X = \mathscr{A}X\mathrm{d}t + F(t,X)\mathrm{d}t + G(t,X)\left(\mathscr{U}^{(i)}(t,X;\theta)\mathrm{d}t + \frac{1}{\sqrt{\rho}}\mathrm{d}W(t)\right), \tag{4.2}$$

where $X(0)$ is an $\mathscr{F}_0-$measurable, $H-$valued random variable, and $\mathscr{A}:D(\mathscr{A})\subset H\to H$ is a linear operator, where $D(\mathscr{A})$ denotes here the domain of $\mathscr{A}$. $F:[0,T]\times H\to H$ and $G:[0,T]\times U\to H$ are nonlinear operators that satisfy properly formulated Lipschitz conditions associated with the existence and uniqueness of solutions to eq. (4.1) as described in [70, Theorem 7.2]. The term $\mathscr{U}^{(i)}(t,X;\theta)$ is a control operator on Hilbert space H parameterized by a finite set of decision variables θ. We view these dynamics in an iterative fashion in order to realize an iterative method. As such, the superscript (i) refers to the iteration number.

In what follows, $\langle\cdot,\cdot\rangle_S$ denotes the inner product in a Hilbert space S and $C([0,T];H)$ denotes the space of continuous processes in H for $t\in[0,T]$. Control tasks defined over SPDEs typically quantify task completion by a measurable functional $J:C([0,T];H)\to\mathbb{R}$ referred to as the cost functional, given by

$$J\big(X(\cdot,\omega)\big) = \phi\big(X(T),T\big) + \int_t^T \ell\big(X(s),s\big)\mathrm{d}s, \tag{4.3}$$

where $X(\cdot,\omega)\in C([0,T];H)$ denotes the entire state trajectory, $\phi\big(X(T),T\big)$ is a terminal state cost and $\ell\big(X(s),s\big)$ is a state cost accumulated over the time horizon $s\in[t,T]$. With

this, we define the terms of eq. (1.1). More information can be found in Appendix B.

Define the *Free energy* of cost function $J(X)$ with respect to uncontrolled path measure $\mathscr{L}$ and temperature $\rho \in \mathbb{R}$ as [57]

$$V(X) := -\frac{1}{\rho} \ln \mathbb{E}_{\mathscr{L}} \Big[\exp\big(-\rho J(X)\big) \Big]. \tag{4.4}$$

Also, the *Generalized Entropy* of controlled path measure $\mathscr{L}^{(i)}$ with respect uncontrolled path measure $\mathscr{L}$ is defined as

$$S\Big(\tilde{\mathscr{L}} || \mathscr{L}\Big) := \begin{cases} -\int_{\Omega} \frac{\mathrm{d}\mathscr{L}^{(i)}}{\mathrm{d}\mathscr{L}} \ln \frac{\mathrm{d}\mathscr{L}^{(i)}}{\mathrm{d}\mathscr{L}} \mathrm{d}\mathscr{L}, \text{if } \mathscr{L}^{(i)} << \mathscr{L}, \\ +\infty, \quad \text{otherwise}, \end{cases} \tag{4.5}$$

where "$<<$" denotes absolute continuity [57].

The relationship between free energy and relative entropy was extended to a Hilbert space formulation in [57]. Based on the free energy and generalized entropy definitions, eq. (1.1) with temperature $T = \frac{1}{\rho}$ becomes the so-called Legendre transformation, and takes the form

$$-\frac{1}{\rho} \ln \mathbb{E}_{\mathscr{L}} \Big[\exp(-\rho J) \Big] \leq \Big[\mathbb{E}_{\mathscr{L}^{(i)}}(J) - \frac{1}{\rho} S\Big(\mathscr{L}^{(i)} \,\Big|\Big|\, \mathscr{L}\Big) \Big], \tag{4.6}$$

with equilibrium probability measure in the form of a Gibbs distribution

$$\mathrm{d}\mathscr{L}^{*} = \frac{\exp(-\rho J)\mathrm{d}\mathscr{L}}{\int_{\Omega} \exp(-\rho J)\mathrm{d}\mathscr{L}}, \tag{4.7}$$

Optimality of $\mathscr{L}^{*}$ is verified in [57]. The statistical physics interpretation of inequality eq. (4.6) is that maximization of entropy results in reduction of the available energy. At the thermodynamic equilibrium the entropy reaches its maximum and $V = E - TS$.

The free energy-relative entropy relation provides an elegant methodology to derive novel algorithms for distributed and boundary control problems of SPDEs. This relation is

also significant in the context of SOC literature, wherein optimality of control solutions rely on fundamental principles of optimality such as Pontryagin Maximum Principle [48] or the Bellman Principle of Optimality [49]. Appendix F shows that by applying a properly defined Feynman-Kac argument, the free energy is equivalent to a value function that satisfies the HJB equation. This connection is valid for general probability measures, including measures defined on path spaces induced by infinite-dimensional stochastic systems.

Our derivation is general in the context of [19], wherein they apply a transformation that is only possible for state-dependent cost functions. The proof given in chapter E is novel for a generic state and time dependent cost to the best knowledge of the authors. The observation that the Legendre transformation in eq. (4.6) is connected to optimality principles from SOC motivates the use of eq. (4.7) for the development of stochastic control algorithms.

Flexibility of this approach is apparent in the context of stochastic boundary control problems, which are theoretically more challenging due to the unbounded nature of the solutions [101, 16]. The HJB theory for these settings is not as mature and results are restricted to simplistic cases [102]. Nonetheless, since eq. (4.6) holds for arbitrary measures, the difficulties of related works are overcome by the proposed ITC approach. Hence, in either the stochastic boundary control or distributed control case the free energy represents a lower bound of a *state cost* plus the associated *control effort*. Despite losing connections to optimality principles in systems with boundary control, our strategy in both distributed and boundary control settings is to optimize the *distance* between our parameterized control policies and the optimal measure in eq. (4.7), so that the lower bound of the total cost can be approached by the controlled system. Specifically, we look for a finite set of decision variables θ^* that yield a Hilbert space control input $\mathscr{U}(\cdot)$ that minimizes the distance to the

optimal path measure

$$\theta^* = \underset{\theta}{\operatorname{argmax}}\, S\big(\mathscr{L}^* || \mathscr{L}^{(i)}\big) \tag{4.8}$$

$$= \underset{\theta}{\operatorname{argmax}} \left[-\int_\Omega \frac{\mathrm{d}\mathscr{L}^*}{\mathrm{d}\mathscr{L}^{(i)}} \ln \frac{\mathrm{d}\mathscr{L}^*}{\mathrm{d}\mathscr{L}^{(i)}} \mathrm{d}\mathscr{L}^{(i)} \right]. \tag{4.9}$$

4.2 Stochastic Optimization in Hilbert Spaces

To optimize eq. (4.8), we apply the chain rule for the Radon-Nikodym (RN) derivative twice [1], which has the form

$$\frac{\mathrm{d}\mathscr{L}^*}{\mathrm{d}\mathscr{L}^{(i)}} = \frac{\mathrm{d}\mathscr{L}^*}{\mathrm{d}\mathscr{L}} \frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\mathscr{L}^{(i)}}. \tag{4.10}$$

Note that the first derivative is given by eq. (4.7) while the second derivative is given by a change of measures, or RN derivative, between control and uncontrolled infinite dimensional stochastic dynamics. This change of measure arises from a version of Girsanov's Theorem, provided with a proof in Appendix C. Under the open-loop parameterization

$$\mathscr{U}(t, \mathbf{x}; \theta) = \sum_{\ell=1}^{N} m_\ell(\mathbf{x}) u_\ell(t) = \mathbf{m}(\mathbf{x})^\top \mathbf{u}(t; \theta), \tag{4.11}$$

Girsanov's theorem yields the following change of measure, or RN derivative, between the two SPDEs

$$\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\mathscr{L}^{(i)}} = \exp\left(-\sqrt{\rho} \int_0^T \mathbf{u}(t)^\top \bar{\mathbf{m}}(t) + \frac{\rho}{2} \int_0^T \mathbf{u}(t)^\top \mathbf{M}\mathbf{u}(t) \mathrm{d}t \right), \tag{4.12}$$

with

$$\bar{\mathbf{m}}(t) := \Big[\langle m_1, \mathrm{d}W(t) \rangle_{U_0}, ..., \langle m_N, \mathrm{d}W(t) \rangle_{U_0} \Big]^\top \in \mathbb{R}^N, \tag{4.13}$$

[1] While this is necessary on the right term for our control update, this is applied to the left term for importance sampling, which enhances algorithmic convergence.

$$\mathbf{M} \in \mathbb{R}^{N\times N}, \quad (\mathbf{M})_{ij} := \langle m_i, m_j \rangle_U, \tag{4.14}$$

where $\mathbf{x} \in \mathscr{D} \subset \mathbb{R}^n$ denotes the localization of actuators in the spatial domain $\mathscr{D}$ of the SPDEs and $m_\ell \in U$ are design functions that specify how actuation is incorporated into the infinite dimensional dynamical system. This parameterization can be used for both open loop trajectory optimization as well as for model predictive control. In our experiments we apply model predictive control through re-optimization, and turn eq. (4.11) into an implicit feedback type control. Optimization using eq. (4.8) with policies that explicitly depend on the stochastic field is also possible and is considered using gradient-based optimization in [63, 103, 64].

To simplify the optimization in eq. (4.8), we further parameterize $\mathbf{u}(t;\theta)$ as a simple measurable function. In this case, the parameters θ consist of all step functions $\{\mathbf{u}_i\}$. With this representation, we arrive at our main result–an importance sampled variational controller of the form

Lemma 4.1. *Consider the controlled SPDE in eq.* (4.2) *and a parameterization of the control as specified by eq.* (4.11) *with* θ *consisting of step functions* $\{\mathbf{u}_i\}$. *The iterative control scheme for solving the stochastic control problem*

$$\mathbf{u}^* = \operatorname{argmax} S(\mathscr{L}^* || \tilde{\mathscr{L}}). \tag{4.15}$$

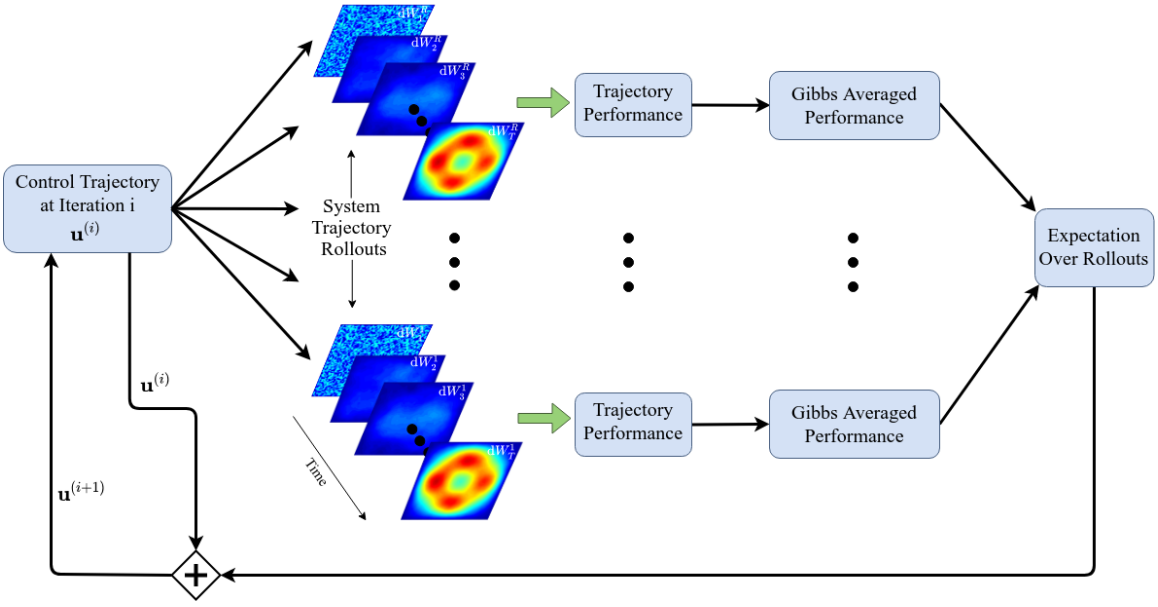


Figure 4.1: Overview of architecture for the control of spatio-temporal stochastic systems, where $\mathrm{d}W_j^r$ denotes a Cylindrical Wiener process at time step j for simulated system rollout r. See eqs. (4.16) and (4.17) and related explanations for a more complete explanation. Although the rollout images appear pictorally similar, they represent different realizations of the noise process $\mathrm{d}W_t$.

is given by the following expression:

$$\mathbf{u}_j{}^{(i+1)} = \mathbf{u}_j{}^{(i)} + \frac{1}{\sqrt{\rho}\Delta t}\mathbf{M}^{-1}\mathbb{E}_{\mathscr{L}^{(i)}}\left[\frac{\exp(-\rho J^{(i)})}{\mathbb{E}_{\mathscr{L}^{(i)}}\left[\exp(-\rho J^{(i)})\right]}\int_{t_j}^{t_{j+1}}\bar{\mathbf{m}}^{(i)}(t)\right], \tag{4.16}$$

$$\textit{where}\quad J^{(i)} := J + \frac{1}{\sqrt{\rho}}\sum_{j=1}^{L}\mathbf{u}_j^{(i)\top}\int_{t_j}^{t_{j+1}}\bar{\mathbf{m}}^{(i)}(t) + \frac{\Delta t}{2}\sum_{j=1}^{L}\mathbf{u}_j^{(i)\top}\mathbf{M}\mathbf{u}_j^{(i)}, \tag{4.17}$$

$$\bar{\mathbf{m}}^{(i)}(t) := \left[\langle m_1, \mathrm{d}W^{(i)}(t)\rangle_U, \ldots, \langle m_N, \mathrm{d}W^{(i)}(t)\rangle_U\right]^\top \in \mathbb{R}^N, \tag{4.18}$$

$$\textit{and}\quad W^{(i)}(t) := W(t) - \sqrt{\rho}\int_0^t \mathscr{U}^{(i)}(s)\mathrm{d}s. \tag{4.19}$$

Proof. See Appendix D. □

4.3 Algorithms for Open Loop and Model Predictive Infinite Dimensional Controllers

The above lemma yields a sampling based iterative scheme for controlling semilinear SPDEs, and is depicted in fig. 4.1. An initial control policy, which is typically initialized by zeros, is

applied to the semilinear SPDE. The controlled SPDE then evolves with different realizations of the Wiener process in a number of trajectory rollouts. The performance of these rollouts is evaluated on the importance sampled cost function in eq. (4.17). These are used to calculate the Gibbs averaged performance weightings $\exp(-\rho J^{(i)})/\mathbb{E}^{(i)}_{\mathscr{L}}[\exp(-\rho J^{(i)}]$. Finally, the outer expectation in eq. (4.16) is evaluated, and used to produce an update to the control policy.

This procedure is repeated over a number of iterations. In the open loop setting, the procedure considers the entire time window $[0,T]$, and the entire control trajectory is optimized in a 'single shot'. In contrast, in the MPC setting a shorter time window $[t_{\text{sim}}, T_{\text{sim}}]$ is considered for I iterations, the control at the current time step $u_I(t_{\text{sim}})$ is applied to the system, and the window recedes backward by a time step Δt. This procedure is described in algorithms 2 and 3.

For the purposes of implementation, we perform the approximation

$$\int_{t_j}^{t_{j+1}} \langle m_l, \mathrm{d}W(t)\rangle_{U_0} \approx \sum_{s=1}^{R} \langle m_l, e_s\rangle_U \Delta\beta_s^{(i)}(t_j), \tag{4.20}$$

where $\Delta\beta_s^{(i)}(t_j)$ are Brownian motions sampled from the zero-mean Gaussian distribution $\Delta\beta_s^{(i)}(t_j) \sim \mathscr{N}(0,\Delta t)$, and $\{e_j\}$ form a complete orthonormal system in U. This is based on truncation of the cylindrical Wiener noise expansion

$$W(t) = \sum_{j=1}^{\infty} \beta_j(t) e_j. \tag{4.21}$$

These algorithms use equations derived in [95] for finite difference approximation of semi-linear SPDEs for Dirichlet and Neumann Boundary conditions. Spatial discretization is done as follows: pick a number of coordinate-wise discretization points J on the coordinate-wise domain $\mathscr{D} = [a,b] \subset \mathbb{R}$ such that each spatial coordinate is discretized as $x_k = a + k\,\frac{b-a}{J}$ where $k = 0, 1, 2, \ldots, J$. For our experiments, the function that specifies how actuation is

Algorithm 2 Open Loop Infinite Dimensional Controller

Function: u = ***OptimizeControl****(Time horizon (T), number of optimization iterations (I), number of trajectory samples per optimization iteration (R), initial field profile (X_0), number of actuators (N), initial control sequences ($u_{T\times N}$) for each actuator, temperature parameter (ρ), time discretization (Δt), actuator centers and variance parameters (θ))*
for $i = 1$ to I **do**
    Initialize $X \leftarrow X_0$
    for $r = 1$ to R **do**
        for $t = 1$ to T **do**
            Sample noise, $\mathrm{d}W(t,x_k) = \sum_{j=1}^{J} \left(e_j(t,x_k)\, \beta_j(t) \right)$, $e_j = \sqrt{2/a}\, sin(j\pi x/a)$ for $x \in L^2(0,a)$
            Compute entries of the actuation matrix $\tilde{M}$ by eq. (4.22)
            Compute the control actions applied to each grid point, $\mathscr{U}(t) = u(t)^T \tilde{M}$
            Propagate the discretized field $X(t)$ [95, Algorighm 10.8]
        end for
        end for
        Compute trajectory cost $J_r^{(i)}$ via eq. (4.17) of the main text
    end for
    end for
    Compute exponential weight of each trajectory $\mathscr{J}_r^{(i)} := \exp\left(-\rho J_r^{(i)}(X) \right)$
    Compute the normalizer $\mathscr{J}_m^{(i)} = \frac{1}{R} \sum_{r=1}^{R} \mathscr{J}_r^{(i)}$
    Update nominal control sequence by eq. (4.16) of the main text
end for
end for
Return: u

Algorithm 3 Model Predictive Infinite Dimensional Controller

Inputs: MPC time horizon (T), number of optimization iterations (I), number of trajectory samples per optimization iteration (R), initial profile (X_0), number of actuators (N), initial control sequences ($u_{T\times N}$) for each actuator, temperature parameter (ρ), time discretization (Δt), actuator centers and variance parameters (θ), total simulation time (T_{sim})
for $t_{\text{sim}} = 1$ to T_{sim} **do**
    $u_I(t_{\text{sim}}) =$ ***OptimizeControl***$(T, I, R, X_0, N, u, \rho, \Delta t, \theta)$
    Apply $u_I(t=1)$ and propagate the discretized field to $t_{\text{sim}} + 1$
    Update the initial field profile $X_0 \leftarrow X(t_{\text{sim}} + 1)$
    Update initial control sequence $u = \left[u_I[2:T,:];\ u_I[T,:] \right]$
end for
end for

implemented by the infinite dimensional control is of the following form:

$$m_l(x_k;\theta) = \exp\left[\frac{-1}{2\sigma_l^2} (x_k - \mu_l)^2 \right], \quad l = 1,\ldots,N \tag{4.22}$$

where, μ_l denotes the spatial position of the actuator on $[a,b]$ and σ_l controls the influence of the actuator on nearby positions.

For MATLAB pseudo-code on sampling space-time noise (step 6 in algorithm 2 and step 7 in algorithm 3), refer to [95, algorithms 10.1 and 10.2]. Note however, that our experiments used cylindrical Wiener noise so $\lambda_j = 1 \; \forall j = 1,\ldots,J$.

We note that the control of SPDEs with cylindrical Wiener noise, as above, can be extended to the case in [19], in which $G(t,X)$ is treated as a trace-class covariance operator $\sqrt{Q}$ of a Q-Wiener process $\mathrm{d}W_Q(t)$. See Appendix H for more details. The resulting iterative control policy is identical to eq. (4.16) derived above.

4.4 Comparisons to Finite-Dimensional Optimization

In light of recent work that apply finite dimensional control after reducing the SPDE model to a set of SDEs or ODEs, we highlight critical advantages of optimizing in Hilbert spaces before discretizating. The main challenge with performing optimization based control after discretization is that SPDEs typically reduce to degenerate diffusion process for which importance sampling schemes are difficult. Consider the finite dimensional SDE representation of eq. (4.1)

$$\mathrm{d}\hat{X} = \mathscr{A}\hat{X}\mathrm{d}t + \mathscr{F}(t,\hat{X})\mathrm{d}t + \mathscr{G}(t,\hat{X})\left(\mathscr{M}\mathbf{u}(t;\theta)\mathrm{d}t + \frac{1}{\sqrt{\rho}}\mathscr{R}\mathrm{d}\beta(t)\right), \tag{4.23}$$

where $\hat{X} \in \mathbb{R}^d$ is a d-dimensional vector comprising of the values of the stochastic field at particular basis elements. The terms $\mathscr{A}$, $\mathscr{F}$, and $\mathscr{G}$ are matrices associated with their respective Hilbert space operators. The matrix $\mathscr{M} \in \mathbb{R}^{d\times k}$, where k is the number of actuators placed in the field. The vector $\mathrm{d}\beta \in \mathbb{R}^m$ collects noise terms and $\mathscr{R}$ collects associated finite dimensional basis vectors of eq. (4.21). The matrix $\mathscr{R} \in \mathbb{R}^{d\times m}$ is composed of d rows, which is the number of basis elements used to spatially discretize the SPDE eq. (4.1), and m columns, which is the number of expansion terms of eq. (4.21) that are used.

Girsanov's theorem for SDEs of the form eq. (4.23) requires the matrix $\mathscr{R}$ to be invertible, as seen in the resulting change of measure, or RN derivative,

$$\begin{aligned}\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\mathscr{L}^{(i)}} = \exp\Bigg(&-\sqrt{\rho}\int_0^T \left\langle \mathscr{R}^{-1}\mathscr{M}\mathbf{u}(s,\theta), \mathrm{d}W(s)\right\rangle_U \\ &+\frac{\rho}{2}\int_0^T \left\langle \mathscr{R}^{-1}\mathscr{M}\mathbf{u}(s,\theta), \mathscr{R}^{-1}\mathscr{M}\mathbf{u}(s,\theta)\right\rangle_U \mathrm{d}s\Bigg)\end{aligned} \tag{4.24}$$

Deriving the optimal control in the finite dimensional space requires that a) the noise term is expanded to at least as many terms as the points on the spatial discretization $d \leq m$, and b) the resulting diffusion matrix $\mathscr{R}$ in eq. (4.23) is full rank. Therefore, increasing finite dimensional approximation accuracy increases the complexity of the sampling process and optimal control computation. This is even more challenging in the case of SPDEs with Q-Wiener noise, where many of the eigenvalues in the expansion of $W(t)$ must be arbitrarily close to zero.

Other finite dimensional approaches as in [104] utilize Gaussian density functions instead of the measure theoretic approach. These approaches are not possible firstly due to the need to define the Gaussian density with respect to a measure other than the Lebesgue measure, which does not exist in infinite dimensions. Secondly, an equivalent Euler-Maruyama time-discretization is not possible without first discretizing spatially. Finally, after spatial discretization, the use of transition probabilities based on density functions requires invertibility of $\mathscr{R}\mathscr{R}^{\mathrm{T}}$ (see Appendix I). These characteristics make Gaussian density based approaches not suitable for deriving optimal control of SPDEs.

4.5 Numerical Results

Performing variational optimization in the infinite dimensional space enables a general framework for controlling general classes of stochastic fields. It also comes with algorithmic benefits from importance sampling and can be applied in either open loop or MPC mode for both boundary and distributed control systems. Critically, it avoids feasibility issues

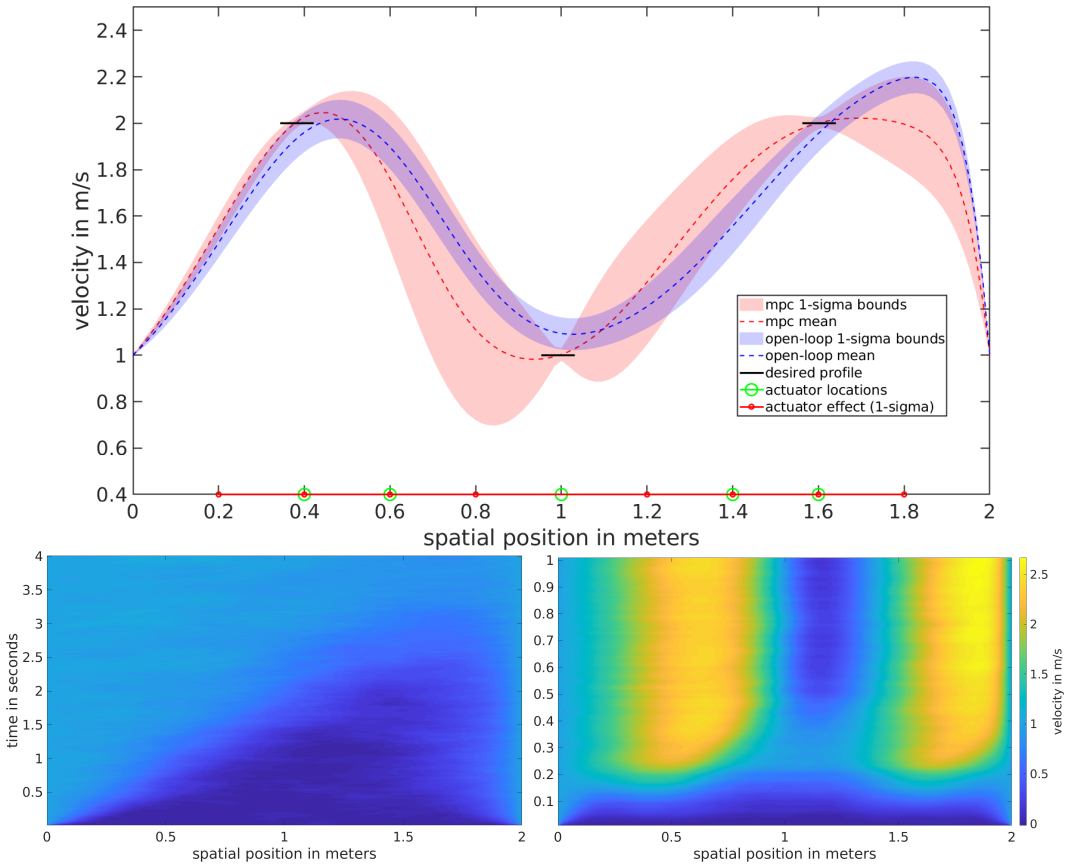


Figure 4.2: Infinite dimensional control of the 1-D Burgers SPDE: (top) Velocity profiles averaged over the 2^{nd}-half of each time horizon over 128 trials. (bottom left) Spatio-temporal evolution of the uncontrolled 1-D Burgers SPDE with Cylindrical Wiener process noise. (bottom right) Spatio-temporal evolution of 1-D Burgers SPDE using MPC.

in optimizing finite dimensional representations of SPDEs. Additional flexibility arises from the freedom to choose the model reduction method that is best suited for the problem without having to change the control update law. Details on the algorithm and more details on each simulated experiment can be found in Appendix J.

4.5.1 Distributed Control of Stochastic PDEs in Fluid Physics

Several simulated experiments were conducted to investigate the efficacy of the proposed control approach. The first explores control of the 1-D stochastic viscous Burgers equation with non-homogeneous Dirichlet boundary conditions. This advection-diffusion equation with random forcing has been studied as a simple model for turbulence [15, 105].

Table 4.1: Summary of Monte Carlo trials for the Stochastic Viscous Burgers Equation

	RMSE			Average σ		
Targets	left	center	right	left	center	right
MPC	0.0344	0.0156	0.0132	0.0309	0.0718	0.0386
Open-loop	0.0820	0.1006	0.0632	0.0846	0.0696	0.0797

The control objective in this experiment is to reach and maintain a desired velocity at specific locations along the spatial domain, depicted in black. In order to achieve the task, the controller must overcome the uncontrolled spatio-temporal evolution governed by an advective and diffusive nature, which produces an apparent velocity wave front that builds across the domain, as depicted on the bottom left of fig. 4.2.

Both open-loop and MPC versions of the control in eq. (4.16) were tested on the 1-D stochastic Burgers equation and the results are depicted in the top subfigure of fig. 4.2. Their performances are compared by averaging the velocity profiles for the 2^{nd}-half of each experiment and repeated over 128 trials. The simulated experiment duration was 1.0 seconds. For the open-loop scheme, 100 optimization iterations with 100 sampled trajectory rollouts per iteration were used. In the MPC setting, 10 optimization iterations were performed at each time step, each using 100 sampled trajectory rollouts.

The results suggest that both the open-loop and MPC schemes have comparable success in controlling the Stochastic Burgers SPDE. The open-loop setting depicts the apparent rightward wavefront that is not as strong in the MPC setting. There is also quite a substantial difference in variance over the trajectory rollouts. The open-loop setting depicts a smaller variance overall, while the MPC setting depicts a variance that shrinks around the objective regions. The MPC performance is desirable since the performance metric only considers the objective regions. The Root Mean Squared Error (RMSE) and variance averaged over the desired regions is provided in table 4.1.

The stochastic Nagumo equation with homogeneous Neumann boundary conditions is a reduced model for wave propagation of the voltage in the axon of a neuron [95]. This SPDE shares a linear diffusion term with the Viscous Burgers equation, as depicted in table 2.1.

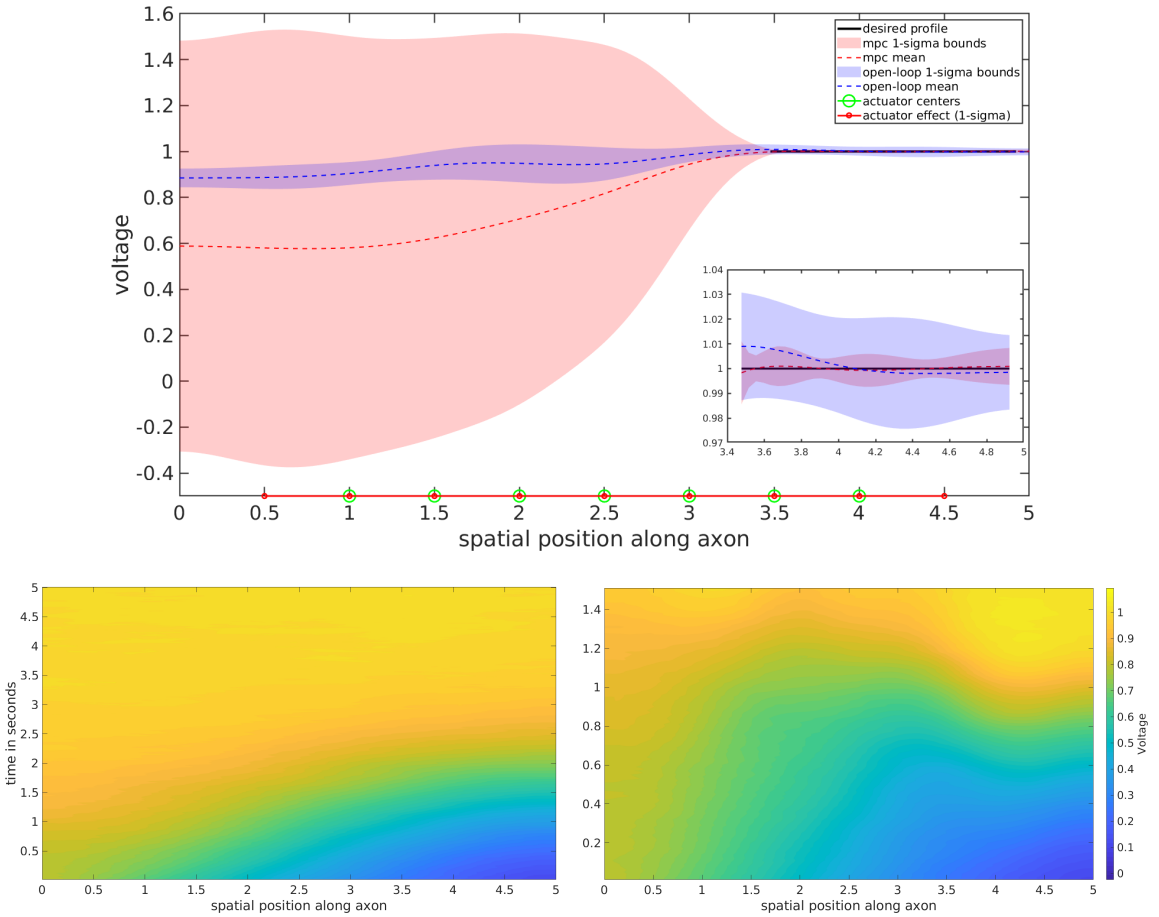


Figure 4.3: Infinite dimensional control of the Nagumo SPDE - Acceleration Task: (top) voltage profiles averaged over the 2$^{\text{nd}}$-half of each time horizon over 128 trials, (bottom left) uncontrolled spatio-temporal evolution for 5.0 seconds, and (bottom right) accelerated activity with MPC within 1.5 seconds.

However, as shown in the bottom left subfigure of fig. 4.3, the nonlinearity produces a substantially different behavior, which propagates the voltage across the axon with our simulation parameters in about 5 seconds. This set of simulated experiments explores two tasks: accelerating the rate at which the voltage propagates across the axon, and suppressing the voltage propagation across the axon. This is analogous to either ensuring the activation of a neuronal signal, or ensuring the neuron remains inactivated.

These tasks are accomplished by reaching either a desired value of 1.0 or 0.0 over the right end of the spatial region for acceleration and suppression, respectively. In both experiments, open-loop and MPC versions of eq. (4.16) were tested, and the results are

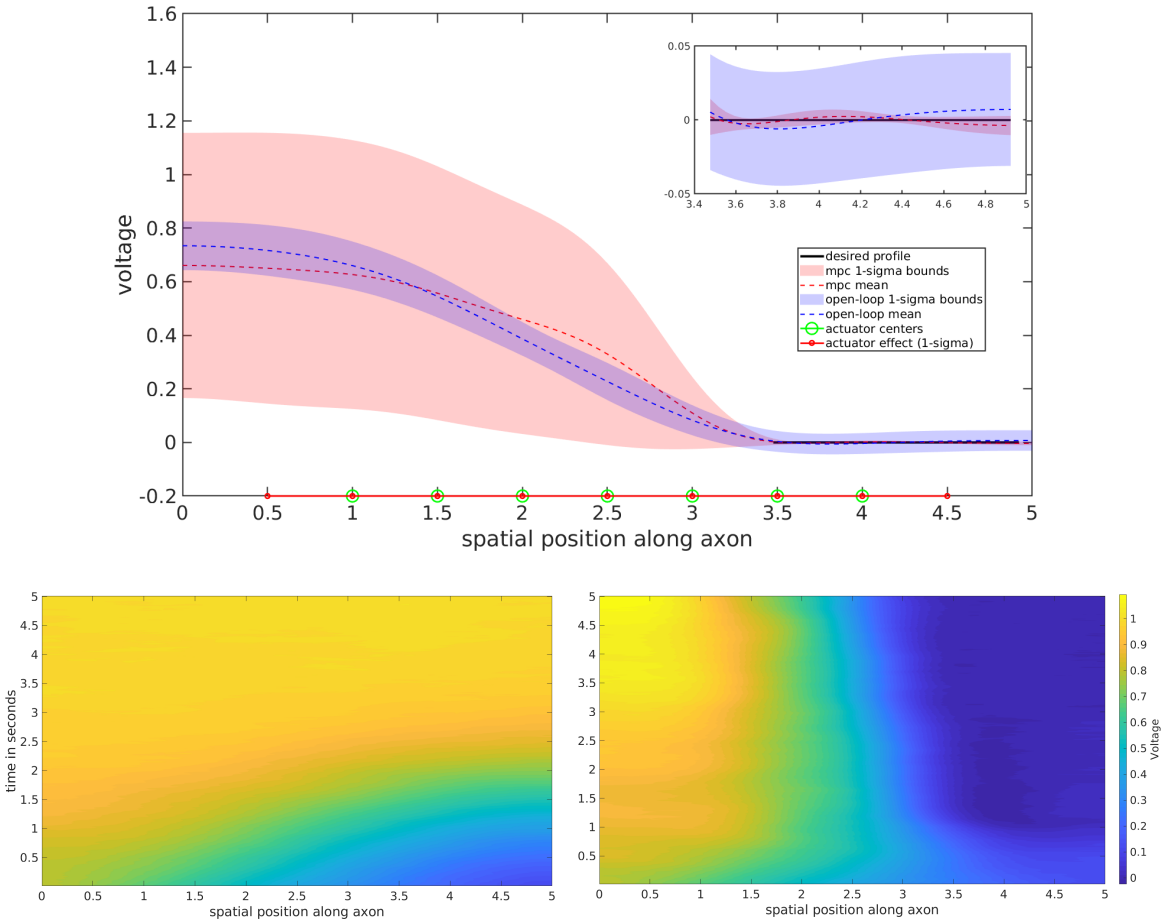


Figure 4.4: Infinite dimensional control of the Nagumo SPDE - Suppression Task: (top) voltage profiles averaged over the 2$^{\text{nd}}$-half of each time horizon over 128 trials, (bottom left) uncontrolled spatio-temporal evolution for 5.0 seconds, and (bottom right) suppressed activity with MPC for 5.0 seconds.

depicted in figs. 4.3 and 4.4. For the open-loop scheme, 200 optimization iterations with 200 sampled trajectory rollouts per iteration were used. In the MPC setting, 10 optimization iterations were performed at each time step, each using 100 sampled trajectory rollouts. State trajectories of both control schemes were compared by averaging the voltage profiles for 2$^{\text{nd}}$-half of each time horizon and repeated over 128 trials.

The results of the two stochastic Nagumo equation tasks suggest that both control schemes achieve success on both the acceleration and suppression tasks. While the performance appears substantially different outside the target region, the two control schemes have very similar performance on the desired region, which is the only penalized region in

Table 4.2: Summary of Monte Carlo trials for Nagumo acceleration and suppression tasks

Task	**Acceleration**		**Suppression**	
Paradigm	MPC	Open-Loop	MPC	Open-Loop
RMSE	$6.605e^{-4}$	0.0042	0.0021	0.0048
Avg. σ	0.0059	0.0197	0.0046	0.0389

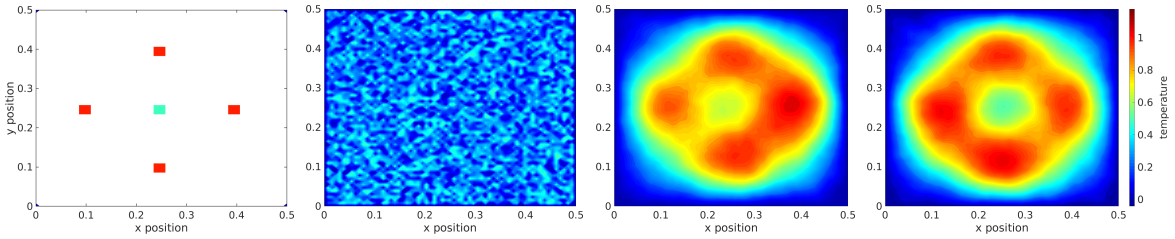


Figure 4.5: Infinite Dimensional control of the 2D-Heat SPDE under homogeneous Dirichlet boundary conditions: (first) desired temperature values at specified spatial regions, (second) random initial temperature profile, (third) temperature profile half way through the experiment and (fourth) temperature profile at the end of experiment.

the optimization objective. In the top subfigures of figs. 4.3 and 4.4, the desired region is zoomed in on. The zoomed in views depict a higher variance in the state trajectories of the open-loop control scheme than the MPC scheme.

As in the stochastic viscous Burgers experiment, there is an apparent trade-off between the two control schemes. The MPC scheme yields a desirable lower variance in the region that is being considered for optimization, but produces state trajectories with very high variance outside the goal region. The open loop control is understood as seeking to achieve the task by reaching low variance trajectories everywhere, while the MPC scheme is understood as acting reactively (i.e. re-optimizes based on state measurements) to a propagating voltage signal. The RMSE and variance averaged over the desired region of 128 trials of each experiment is given in table 4.2.

The next simulated experiment explores scalability to 2D spatial domains by considering the 2D stochastic heat equation with homogeneous Dirichlet boundary conditions. This experiment can be thought of as attempting to heat an insulated metal plate to specified temperatures in specified regions while the edges remain at a temperature of 0 in some scale. The desired temperatures and regions associated with this experiment are depicted in the

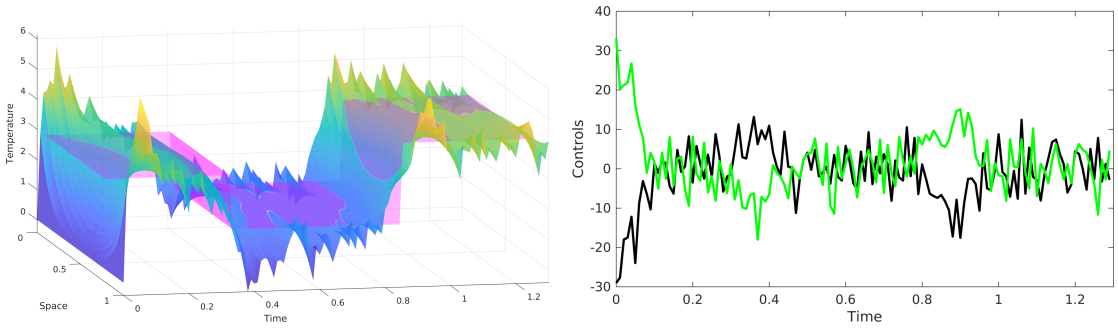


Figure 4.6: Boundary control of stochastic 1-D heat equation: (left) Temperature profile over the 1D spatial domain over time. The magenta surface corresponds to the spatio-temporal desired temperature profile. Colors that are more red correspond to higher temperatures, and colors that are more violet correspond to lower temperature. (right) Control inputs at the left boundary in black and the right boundary in green entering through Neumann boundary conditions.

left subfigure of fig. 4.5. This experiment tests the MPC scheme.

Starting from a random initial temperature profile as in the second subfigure of fig. 4.5, and using a time horizon of 1.0 seconds, the MPC controller is able to achieve the desired temperature profile towards the end of the time horizon as shown in the fourth subfigure of fig. 4.5. The third subfigure of fig. 4.5 depicts the middle of the time horizon. The MPC controller used 5 optimization iterations at every timestep and 25 sampled trajectories per iteration.

This result suggests that in this case, this approach can handle the added complexity of 2D stochastic fields. As depicted in the right subfigure of fig. 4.5, the proposed MPC control scheme solves the task of reaching the desired temperature at the specified spatial regions.

4.5.2 Boundary Control of Stochastic PDEs

The control update in eq. (4.16) describes control of SPDEs by distributing actuators throughout the field. However, our framework can also handle systems with control and noise at the boundary. A key requirement is to write such dynamical systems in the *mild*

form

$$
\begin{aligned}
X(t) = e^{t\mathscr{A}}\xi + \int_0^t e^{(t-s)\mathscr{A}} F_1(t,X)\mathrm{d}s \\
+ (\lambda I - \mathscr{A})\Bigg[\int_0^t e^{(t-s)\mathscr{A}} \mathsf{D}\big(F_2(t,X) + G(t,X)\mathscr{U}(t,X;\theta)\big)\mathrm{d}s \\
+ \int_0^t e^{(t-s)\mathscr{A}} \mathsf{D}B(t,X)\mathrm{d}V(s)\Bigg], \quad \mathbb{P}\ a.s.
\end{aligned} \tag{4.25}
$$

where the operator D corresponds to the boundary conditions of the problem, and is called the *Dirichlet map* (*Neumann map*, resp.) for Dirichlet (Neumann, resp.) boundary control/noise. These maps take operators defined on the boundary Hilbert space Λ_0 to the Hilbert space H of the domain. λ is a real number also associated with the boundary conditions. The operator $\mathrm{d}V$ describes a cylindrical Wiener process on the boundary Hilbert space Λ_0. For further details, the reader can refer to the discussion in [16, Section 2.5 & Appendix C.5] and in Appendix G.

Studying optimal control problems with dynamics as in eq. (4.25) is rather challenging. Therein HJB theory requires additional regularity conditions and proving convergence of eq. (4.25) becomes nontrivial, especially when considering Dirichlet boundary noise. Numerical results are limited to simplistic problems. Nevertheless, eq. (4.16) is extended to the case of boundary control by similarly using tools from Girsanov's theorem to obtain the change of measures, or RN derivative

$$
\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\mathscr{L}^{(i)}} = \exp\bigg(-\int_0^T \big\langle B^{-1}G\mathscr{U}, \mathrm{d}V(s)\big\rangle_{\Lambda_0} + \frac{1}{2}\int_0^T ||B^{-1}G\mathscr{U}||^2_{\Lambda_0}\mathrm{d}s\bigg), \tag{4.26}
$$

which was was also utilized in reference [106] for studying solutions of SPDEs similar to eq. (4.25). Using the control parameterization of the distributed case above results in the same approach described in eq. (4.16) with inner products taken with respect to the boundary Hilbert space Λ_0 to solve stochastic boundary control problems.

The stochastic 1-D heat equation under Neumann boundary conditions was explored

to conduct simulated experiments that investigate the efficacy of the proposed framework in stochastic boundary control settings. The objective is to track a time-varying profile that is uniform in space by actuation only at the boundary points. The MPC scheme of eq. (4.16), with 10 optimization iterations per time step is depicted in the left subfigure of fig. 4.6. The random sample of the controlled state trajectory, depicted in a violet to red color spectrum, remains close to the time-varying desired profile, depicted in magenta. The associated bounded actuation signals acting on the two boundary actuators are depicted in the right subfigure of fig. 4.6.

As suggested by the results of the simulated experiments, the authors note a clear empirical iterative improvement of the control policy on each of the experiments. This necessitates a deeper theoretical analysis of the convergence of the proposed algorithm, and is influenced by several of the parameters that appear in algorithms 2 and 3. The parameter ρ, which appears in the controlled and controlled dynamics eqs. (4.1) and (4.2) and also appears in the Legendre transformation eq. (4.6), influences the intensity of the stochasticity and the relative weightings of the terms in eq. (4.17), which in general leads to an exploration-exploitation trade off. The number of rollouts also has a significant effect on the empirical performance. In general, a larger number of rollouts is advantageous due to a more representative sampling of state space, as well as a better approximation to the expectation, yet can lead to a larger computational burden. In the MPC setting, the time horizon has a significant effect on the empirical performance. This is typical to MPC methods as a short receding window can cause the algorithm to be myopic, while a large receding window recovers the 'single shot' or open loop performance. Finally the spatial and temporal discretization size has a significant effect on algorithmic performance due to the errors introduced in large spatial or temporal steps in the resulting discrete equations, which may ultimately fail the Courant–Friedrichs–Lewy conditions of the SPDE.

The above experiments were designed to cover stochastic SPDEs with nonlinear dynamics, multiple spatial dimensions, time-varying objectives, and systems with both distributed

and boundary actuation. This range explores the versatility of the proposed framework to problems of many different types. Throughout these experiments, the control architecture produces state trajectories that solve the objective with high probability for the given stochasticity.

4.6 Conclusion

This manuscript presented a variational optimization framework for distributed and boundary control of stochastic fields based on the free energy-relative entropy relation. The approach leverages the inherent stochasticity in the dynamics for control, and is valid for generic classes of infinite-dimensional diffusion processes. Based on thermodynamic notions that have demonstrated connections to established SOC principles, algorithms were developed that bridge the gap between abstract theory and computational control of SPDEs. The distributed and boundary control experiments demonstrate that this approach can successfully control complex physical systems in a variety of domains.

This research opens new research directions in the area of control of stochastic fields that are ubiquitous in domains of physics. Based on the use of forward sampling, future research on the algorithmic side will include the development of efficient methods for representation and propagation of stochastic fields using techniques in machine learning such as Deep Neural Networks. Other directions include explicit feedback parameterizations and, in the context of boundary control, HJB approaches in the ITC formulation.

CHAPTER 5

VARIATIONAL OPTIMIZATION BASED REINFORCEMENT LEARNING FOR INFINITE DIMENSIONAL STOCHASTIC SYSTEMS

In contrast to the open loop and MPC approach developed in chapter 4, here we parametrize the policy as having explicit dependence on the state, and thereby consider the problem of optimizing an explicit feedback controller. We again formulate the optimization problem from the ITC principles considered in chapter 4, which together enable a middle ground between recent results in DL and traditional stochastic optimal control: We approach SPDEs with infinite dimensional stochastic calculus, yet apply highly successful DL techniques. We develop a new method fusing together variational optimization, episodic reinforcement learning, and measure theoretic stochastic calculus in infinite dimensions.

This chapter views the optimization problem through the lens of reinforcement learning in Hilbert spaces, and develops a measure theoretic loss function, which is optimized by widely successful DL techniques in order to episodically train a parametrized control policy which is nonlinear in the system state. The resulting algorithm, called IDVRL, incorporates explicit feedback of the entire SPDE and allows for arbitrary non-linear policies such as Feed-forward Neural Networks (FNNs), CNNs and RNNs.

Furthermore, we develop novel techniques to handle numerical integration of of policy networks over spatial domains which we call *SparseForwardPass* for FNN and CNN policies. This increases the numerical efficiency of the overall approach, enabling scalability to 2D and 3D problems. The IDVRL algorithm is applied to several 1D systems in fluid dynamics, and demonstrates effectiveness of the approach. Additionally, IDVRL is applied to a 2D system to demonstrate scalability of the approach.

Since the algorithm is derived in infinite-dimensional space, any choice of numerical approximation scheme such as finite difference, spectral Galerkin or finite-element can be

used to approximate trajectory samples. In addition, as a result of performing optimization in infinite dimensional space, the derivation is valid for the stochastic versions of all PDEs included in table 2.1 and therefore is general.

5.1 Problem Formulation

This work proposes control of a large class of infinite-dimensional systems described by SPDEs that are of *semi-linear* form. There are other ways to express such systems, however here we take the approach of expressing the system as evolving on time-indexed separable Hilbert spaces in order to leverage several mathematical tools developed in such spaces. Consider the general semi-linear controlled SPDE given by

$$\mathrm{d}X = \big(\mathscr{A}X + F(t,X)\big)\mathrm{d}t + G(t,X)\big(\Phi(t,X,\mathbf{x};\Theta^{(k)})\mathrm{d}t + \frac{1}{\sqrt{\rho}}\mathrm{d}W(t)\big), \tag{5.1}$$

where $X(t) \in \mathscr{H}$ is the state of the system which evolves on the Hilbert space $\mathscr{H}$, the linear and nonlinear operators $\mathscr{A} : \mathscr{H} \to \mathscr{H}$ and $F(t,X) : \mathbb{R} \times \mathscr{H} \to \mathscr{H}$ (resp.) are uncontrolled drift terms, $\Phi(t,X,\mathbf{x};\Theta^{(k)}) : \mathbb{R} \times \mathscr{H} \times \mathbb{R}^3 \to \mathscr{H}$ is the nonlinear control policy paramterized by $\Theta^{(k)}$ at the k^{th} iteration, $\mathrm{d}W(t) : \mathbb{R} \to \mathscr{H}$ is a Cylindrical spatio-temporal noise process (i.e. space-time white noise), and $G(t,X)$ is nonlinearity that affects both the Cylindrical noise and the control. It is used to incorporate the effects of actuation on either the field (distributed) or at the boundaries. Referring back to table 2.1, the generality of the *Hilbert spaces formulation* becomes clear as any semi-linear PDE can be handled by appropriately choosing $\mathscr{A}$ and F. For a more complete introduction, including some mild but necessary assumptions and clear definitions of the Cylindrical process, see Appendix A and the references therein.

Define the uncontrolled and controlled probability measures associated with eq. (5.1) as $\mathscr{L}$ and $\tilde{\mathscr{L}}$, respectively. These measures roughly describe the probabilistic evolution of the system, with the probability density function as a finite dimensional analog. In this case,

eq. (1.1) takes the form [107]

$$-\frac{1}{\rho}\log\mathbb{E}_{\mathscr{L}}\Big[\exp(-\rho J)\Big]=\min_{\Phi}\Big[\mathbb{E}_{\tilde{\mathscr{L}}}(J)+\frac{1}{\rho}D_{KL}(\tilde{\mathscr{L}}\,\big|\big|\,\mathscr{L})\Big], \tag{5.2}$$

where $J=J(X)$ can be viewed as an arbitrary state cost function. The associated "Work" and "Entropy" terms that minimize this expression describe a minimum "energy"[1] measure. Sampling from this measure would simultaneously minimize state cost and the Kullback-Leibler (KL)-divergence between the controlled and uncontrolled distributions, which in this case is roughly interpreted as control effort. The measure that optimizes eq. (5.2) is the so-called Gibbs measure

$$\mathrm{d}\mathscr{L}^*=\frac{\exp(-\rho J)\mathrm{d}\mathscr{L}}{\mathbb{E}_{\mathscr{L}}\big[\exp(-\rho J)\big]}. \tag{5.3}$$

While it is not known how to sample directly from eq. (5.3), the goal of variational optimization methods is to incrementally reduce the distance (defined in the KL divergence sense) between the controlled distribution $\tilde{\mathscr{L}}$ and the optimal measure eq. (5.3). We formulate our variational minimization problem as

$$\begin{aligned}\Theta^*&=\underset{\Theta}{\operatorname{argmin}}\,D_{KL}(\mathscr{L}^*||\tilde{\mathscr{L}})\\&=\underset{\Theta}{\operatorname{argmin}}\left[\int\log\Big(\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}}\Big)\frac{\mathrm{d}\mathscr{L}^*}{\mathrm{d}\mathscr{L}}\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}}\mathrm{d}\tilde{\mathscr{L}}\right]=\underset{\Theta}{\operatorname{argmin}}\;L\end{aligned} \tag{5.4}$$

A more detailed derivation can be found in the Appendix K. Finally, we introduce a version of Girsanov's theorem (found in Appendix C) between the uncontrolled and controlled processes, resulting in the change of measures, or RN derivative, given as

$$\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}}=\exp\left\{-\sqrt{\rho}\int_0^T\Big\langle\Phi(t,X,\mathbf{x};\Theta^{(k)}),\mathrm{d}W(t)\Big\rangle-\rho\frac{1}{2}\int_0^T\Big\langle\Phi(t,X,\mathbf{x};\Theta^{(k)}),\Phi(t,X,\mathbf{x};\Theta^{(k)})\Big\rangle\mathrm{d}t\right\}. \tag{5.5}$$

[1] The term energy here is used loosely to describe the landscape for work and entropy

Plugging in eq. (5.3) and eq. (5.5) (for importance sampling), the loss-function L becomes

$$L = \mathbb{E}_{\tilde{\mathscr{L}}}\left[\underbrace{\frac{\exp(-\rho\tilde{J})}{\mathbb{E}_{\tilde{\mathscr{L}}}\left[\exp(-\rho\tilde{J})\right]}}_{ImportanceWeight}\left(-\sqrt{\rho}\int_0^T \underbrace{\left\langle \Phi(t,X,\mathbf{x};\Theta^{(k)}), \mathrm{d}W(t)\right\rangle}_{NoiseInnerProduct}\right.\right.$$
$$\left.\left. -\frac{1}{2}\rho\int_0^T \underbrace{\left\langle \Phi(t,X,\mathbf{x};\Theta^{(k)}), \Phi(t,X,\mathbf{x};\Theta^{(k)})\right\rangle}_{PolicyInnerProduct}\mathrm{d}t\right)\right], \tag{5.6}$$

where $\tilde{J}$ is defined by

$$\tilde{J} = \underbrace{J}_{StateCost} + \frac{1}{\sqrt{\rho}}\int_0^T \underbrace{\left\langle \Phi(t,X,\mathbf{x};\Theta^{(k)}), \mathrm{d}W(t)\right\rangle}_{NoiseInnerProduct} + \frac{1}{2}\int_0^T \underbrace{\left\langle \Phi(t,X,\mathbf{x};\Theta^{(k)}), \Phi(t,X,\mathbf{x};\Theta^{(k)})\right\rangle}_{PolicyInnerProduct}\mathrm{d}t. \tag{5.7}$$

The intermediate steps that lead to the above final forms of eq. (5.4) and eq. (5.6) can be found in Appendix K. The loss-function L exponentiates the cost of the system trajectories, evaluated by $\tilde{J}$, to produce a weighted average of the mixed control-noise term and the quadratic control term. We minimize this loss via Stochastic Gradient Descent (SGD). The resulting Variational RL with learn rate γ is an incremental update of the form

$$\Theta^{(k+1)} = \Theta^{(k)} - \gamma\nabla_{\Theta}L. \tag{5.8}$$

We contrast this work to prior work that also use variational optimization to approximate optimal probability measures, as in [108]. There, the authors obtain a time-varying policy of step-functions that results in parameter update-rules requiring inversion of a jacobian. Our proposed approach instead uses an arbitrary non-linear feedback policy and produces a SGD-based minimization that can leverage well-known backprop-based algorithms such as ADA-Grad [109] and ADAM [110].

Although the state may be described by an infinite-dimensional vector in a Hilbert space,

many physical realizations of actuation are defined on finite subspaces. The above derivation keeps Φ as mapping into the Hilbert space, insinuating that the actuation may be distributed everywhere and infinite-dimensional. However, the goal of this work is to ultimately use finite-action policy networks to control eq. (5.1). As such, we refine Φ as

$$\Phi(t, X, \mathbf{x}; \Theta^{(k)}) = \mathbf{m}(\mathbf{x})^\top \varphi(X; \Theta^{(k)}), \tag{5.9}$$

where $\varphi(X; \Theta^{(k)}) : \mathscr{H} \to \mathbb{R}^N$ is a finite policy network with N control outputs representing N distributed (or boundary) actuators. The function $\mathbf{m}(\mathbf{x}) : D \to \mathbb{R}^N \times \mathscr{H}$ represents the effect of the finite actuation on the infinite-dimensional field, where D is the domain of the finite spatial region. Some examples of $\mathbf{m}(\mathbf{x})$ are Gaussians-like exponential functions with mean centered at the actuator location (for distributed control) and indicator functions (for boundary control).

5.2 Algorithm and Network Architecture

The above derivation provides a mathematical framework for updating the weights of a policy network in a RL setting. In order to implement it as an algorithm, data must be generated either from a physics-based or data-based model, or from interactions with a real system. Notice that since the only term from the dynamics to appear in eqs. (5.6) and (5.7) is the Cylindrical noise term dW, there is no need to have an explicit SPDE model. As a result, any black-box methods that incorporate spatio-temporal stochasticity can be used to generate sample trajectories of the system.

The above derivation introduces a unique problem for our proposed reinforcement learning framework that has not been addressed in prior work. Each inner product in Hilbert space in eqs. (5.6) and (5.7) represents a spatial integration over a finite region D. To the knowledge of the authors, integration over a policy network has not been attempted to date. However in this work, we integrate spatially over the input to the network. Consider the inner product indicated as *PolicyInnerProduct*. The representation of this inner product as a

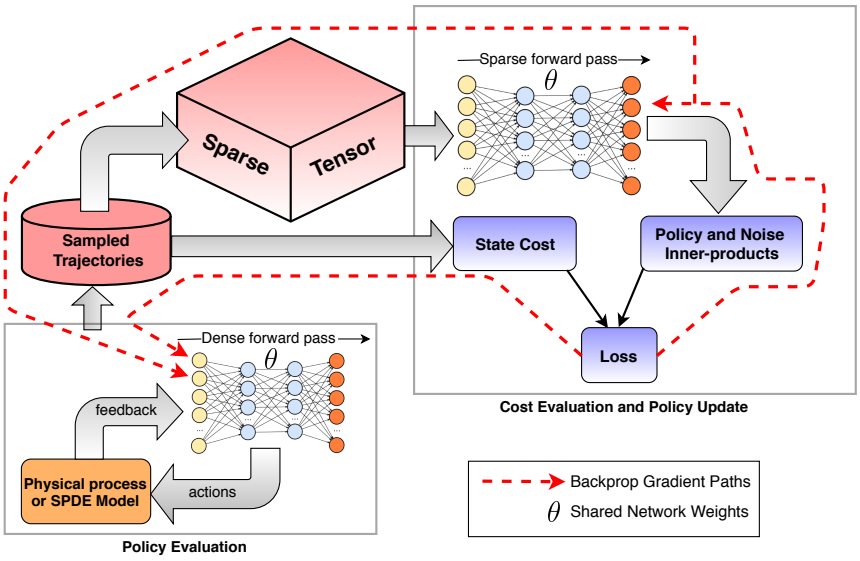


Figure 5.1: Block diagram of computational graph for the IDVRL algorithm.

spatial integration takes the form

$$\int_0^T \left\langle \Phi(X,\mathbf{x};\Theta^{(k)}), \Phi(X,\mathbf{x};\Theta^{(k)}) \right\rangle \mathrm{d}t = \int_0^T \iint_D \varphi(X(t,x,y);\Theta^{(k)})^\top M(x,y)\varphi(X(t,x,y);\Theta^{(k)})\mathrm{d}x\mathrm{d}y\mathrm{d}t$$
$$= \int_0^T \sum_{j=1}^{\infty} \varphi(X(e_j);\Theta^{(k)})^\top M(e_j)\varphi(X(e_j);\Theta^{(k)})\mathrm{d}t, \qquad (5.10)$$

where $D \subseteq \mathbb{R}^2$ is the problem domain, $\{e_j \in \mathscr{H} : j = 0,1,2,\dots\}$ forms an orthonormal basis over $\mathscr{H}$, and $M(\mathbf{x}) = \mathbf{m}(\mathbf{x})\mathbf{m}(\mathbf{x})^\top$. After discretization on a 2D grid, the basis becomes a finite set $\{e_j \in \mathbb{R}^{J^2} : j = 0,1,2,\dots\}$, where each element is a one-hot vector. Thus, evaluating the spatial integral is reduced to summing up forward passes through the policy network with each pixel considered individually. Note that this spatial integration approach is agnostic to choice of discretization scheme.

Spatially integrating over the policy network is a memory intensive task, where the storage becomes (J^2, J, J) for each sample over the time horizon. However, given that the basis elements of each (J,J) "image" have only one activated "pixel", the resulting tensor is tremendously sparse. As such, each layer's activation can be computed with a sparse matrix multiplication, resulting in what we call a *SparseForwardPass* method that is not memory intensive for relatively large 2D problems. This can be applied to both FNNs and

CNNs. For CNNs, activation can be achieved by matrix multiplication with a Toeplitz matrix constructed from the filter coefficients [111].

A summary of our architecture is depicted in fig. 5.1. A policy network with initialized weights is passed through a model or physical realization of the system to produce state trajectories, which are used to compute a state cost as well as a sparse tensor that is used to compute the inner products in eqs. (5.6) and (5.7) in a memory and time-efficient manner. Finally the loss is computed and passed to a gradient-based optimization algorithm. This approach is independent of specific policy network architecture used, which can often be problem dependent. In this work we used two different networks: a FNN for 1D SPDE and a CNN for 2D SPDE.

The resulting IDVRL algorithm is shown in algorithm 4, wherein subscript implies an element of the corresponding vector. The input terms are time horizon (T), number of iterations (K), number of rollouts (R), initial state (X_0), number of actuators (N), noise variance (ρ), time discretization (Δt), actuator locations (μ), actuator variance (σ_μ, for distributed control cases), and initial network parameters ($\Theta^{(0)}$). We note that the function $GradientOptimize(L, \Theta^{(k)})$ represents the update from eq. (5.8). As mentioned above, this is handled by any variant of SGD, which performs backpropagation through the network. The computational graph of the proposed algorithm has multiple backprop paths, as shown by the dotted red line in fig. 5.1. For more information on $SampleNoise()$, refer to [95, Chapter 10].

5.3 Simulation Results and Discussion

We applied the IDVRL algorithm to reaching tasks for several SPDEs in simulation in both distributed and boundary control settings. In each reaching task, the policy has to control the system to achieve a desired profile in certain parts of the spatial domain. These simulated experiments were developed via computational graphs implemented in Tensorflow [112] to leverage GPU parallelization for training as well as sparse linear algebra operations

Algorithm 4 Infinite Dimensional Variational Reinforcement Learning

Function: $\Theta^* = \textbf{\textit{OptimizePolicyNetwork}}(T, K, R, X_0, N, \rho, \Delta t, \mu, \sigma_\mu, \Theta^{(0)})$
Compute $\mathbf{m}(\mathbf{x}), M(\mathbf{x}) \; \forall \; \mathbf{x} \in D$
for $k = 1$ to K **do**
  for $r = 1$ to R **do**
    for $t = 1$ to T **do**
      $\mathrm{d}W_t \leftarrow SampleNoise()$
      $X_t \leftarrow Propagate(X_{t-1}, \Theta^{(k)}, \mathrm{d}W_t)$ via eq. (5.1)
      $J_r \leftarrow J_r + StateCost(X_t)$
      $S_t \leftarrow SparseForwardPass(\Theta^{(k)}, X_t)$
      $N_t \leftarrow NoiseInnerProduct\big(S_t, \mathrm{d}W_t, \mathbf{m}(\mathbf{x})\big)$
      $P_t \leftarrow PolicyInnerProduct\big(S_t, M(\mathbf{x})\big)$
    end for
    $P,\ N \leftarrow Sum(P_t),\ Sum(N_t)$
    $\tilde{J}_r \leftarrow \tilde{J}(P, N, J_r)$
  end for
  $W \leftarrow ImportanceWeight(\tilde{J})$
  $L \leftarrow ComputeLoss(P, N, W)$ via eq. (5.7)
  $\Theta^{(k+1)} \leftarrow GradientOptimize(L, \Theta^{(k)})$
end for

for *SparseForwardPass*. The data for training the policies was generated by simulating the SPDEs using centered finite-difference approximation for the spatial derivatives on a 1D or 2D grid and a semi-implicit Euler scheme for discretization of the time derivatives. For detailed explanation on these schemes, we refer the reader to [95, Chapters 3 and 10]. For 1D simulations, we used an Alienware laptop with an Intel Core i9-8950HK CPU @ 2.9GHz×12, 32 GB RAM and a NVIDIA GeForce GTX 1080 graphics card. On average, Tensorflow-GPU required around 16 minutes of training time for 1000 iterations. For the 2D simulation, we used Tensorflow-CPU, due to insufficiency of VRAM, which required around 12 hours of training time for 1000 iterations. The code and videos for these experiments are available online [2].

Figure 5.2 (a) and (d) depict the results of the IDVRL algorithm on a task of controlling the 1D heat SPDE with homogeneous Dirichlet boundary conditions. The goal of the task is to raise and maintain the temperature to $T = 1$ at regions around $x = 0.2$ and $x = 0.8$, and

[2] Code: https://github.gatech.edu/eevans41/spde_explicit_feedback_RL, Video: https://youtu.be/6tmky59xhp4

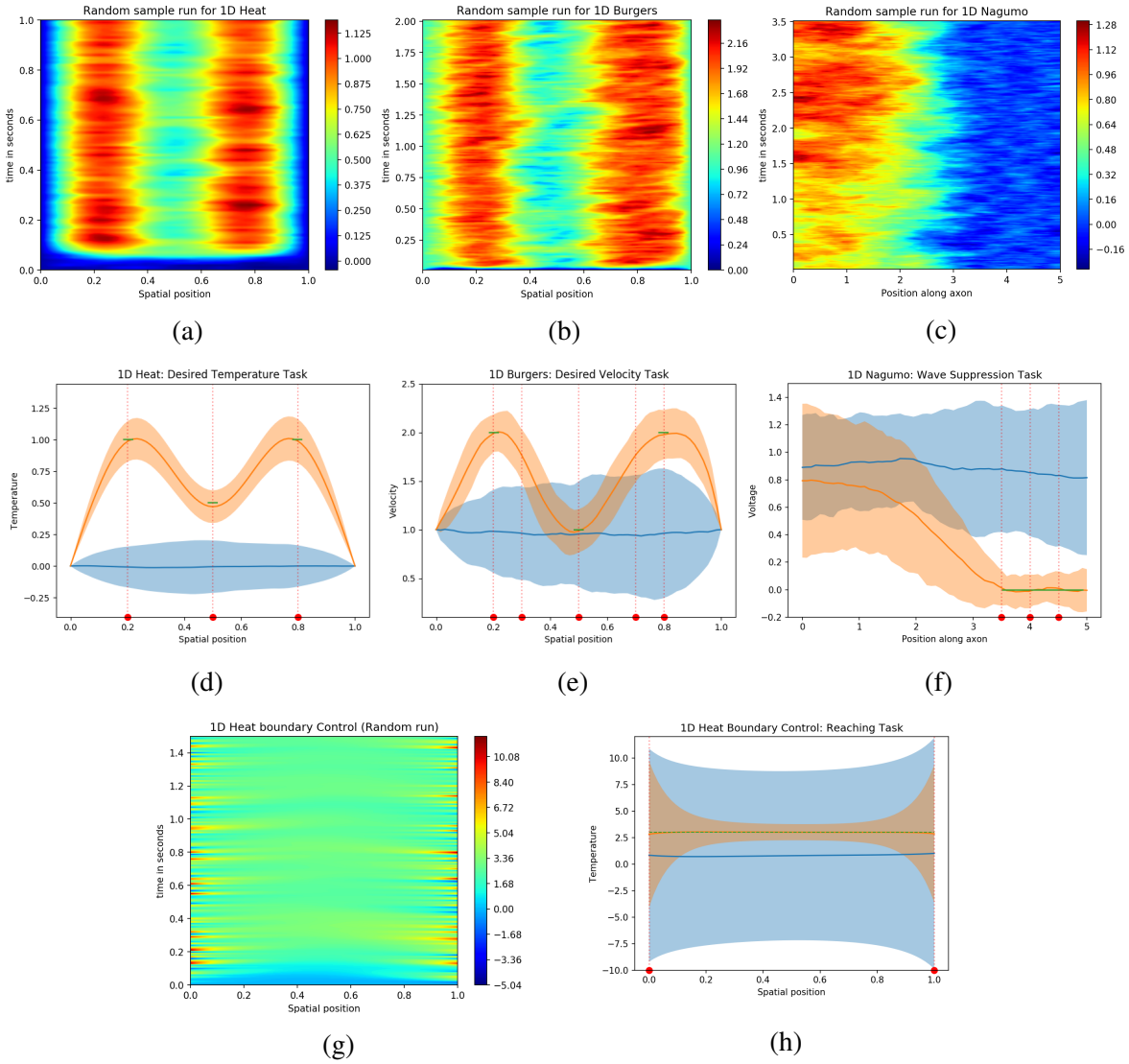


Figure 5.2: Control of 1D SPDEs. (a), (d), (g), (h) correspond to the Heat SPDE, (b), (e) to Burgers SPDE, and (c), (f) to Nagumo SPDE. In (d), (e), (f), (h) blue represents *mean uncontrolled profiles*, orange represents *mean controlled profiles* using the trained policy network, green represents *desired values* in certain spatial regions, and red represents *locations of actuator centers*. The mean and variance statistics are gathered over 200 rollouts. (a), (b), (c), (g) depict a randomly selected trial run to emphasize the presence of spatio-temporal stochasticity. (a-f) depict results for distributed control of SPDEs and (g-h) depict results for boundary control of a SPDE.

$T = 0.5$ at a region around $x = 0.5$. Figure 5.2a) shows the temperature contours of a single realization of the completed task and fig. 5.2d) shows the mean controlled and uncontrolled trajectories at the final time with a 2-σ variance shaded in the corresponding color. The boundary conditions fixed the endpoints to a temperature of $T = 0$, as shown.

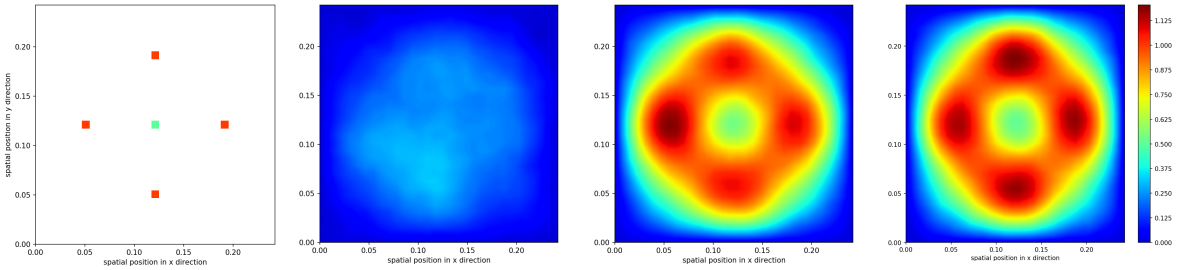
Figure 5.2, (b) and (e) depict the results of the IDVRL algorithm on the task of controlling

the 1D Burgers SPDE with non-homogeneous Dirichlet boundary conditions. In this task the goal is to reach a desired velocity in the medium at given locations. This is challenging given the nonlinear advection behavior of the system in addition to the pure diffusion behavior shown in the 1D heat SPDE task. The advection-diffusion creates an apparent rightwards wave-front that must be accounted for by the policy network in order to achieve the task. Given the increased difficulty of the problem, we added actuators, as indicated by vertical red dotted lines. Despite the added actuators, the task remains severely under-actuated.

Figure 5.2, (c) and (f) depict the IDVRL algorithm on the task of controlling the 1D Nagumo SPDE with homogeneous Neumann boundary conditions. As noted earlier, the Nagumo SPDE represents voltage travelling across the axon of a neuron in the brain. The goal of this task is to suppress the voltage from travelling across the axon. Voltage near 1.0 indicates the voltage has travelled across, and in this suppression task, we seek to keep the voltage at the right end of the axon at $V = 0$. As shown in table 2.1, the Nagumo SPDE has a 3rd order nonlinearity. For this task, we supplied the system with only three actuators near the right end, where voltage must be suppressed.

For the next task, we scaled the IDVRL algorithm to two-dimensional problems. With this task we attempt to control the 2D Heat SPDE with homogeneous Dirichlet boundary conditions with a CNN policy network. The goal of this task it to raise the temperature in five regions. The desired temperature at the four outer regions is $T = 1$ and the desired temperature at the center region is $T = 0.5$. Figure 5.3 depicts a single realization of the controlled task under a significant amount of noise with five actuators.

In contrast to the previous tasks where actuators are distributed in the field, fig. 5.2h depicts a *boundary* control task, where the actuator controls the boundary condition. The RN derivative exists for the case of boundary control of semi-linear SPDEs with boundary noise [15], and we demonstrate that our method similarly extends to this case. The task here is similar to the first case, where the policy network is tasked with reaching a desired value of $T = 3$.



(a) (b) (c) (d)

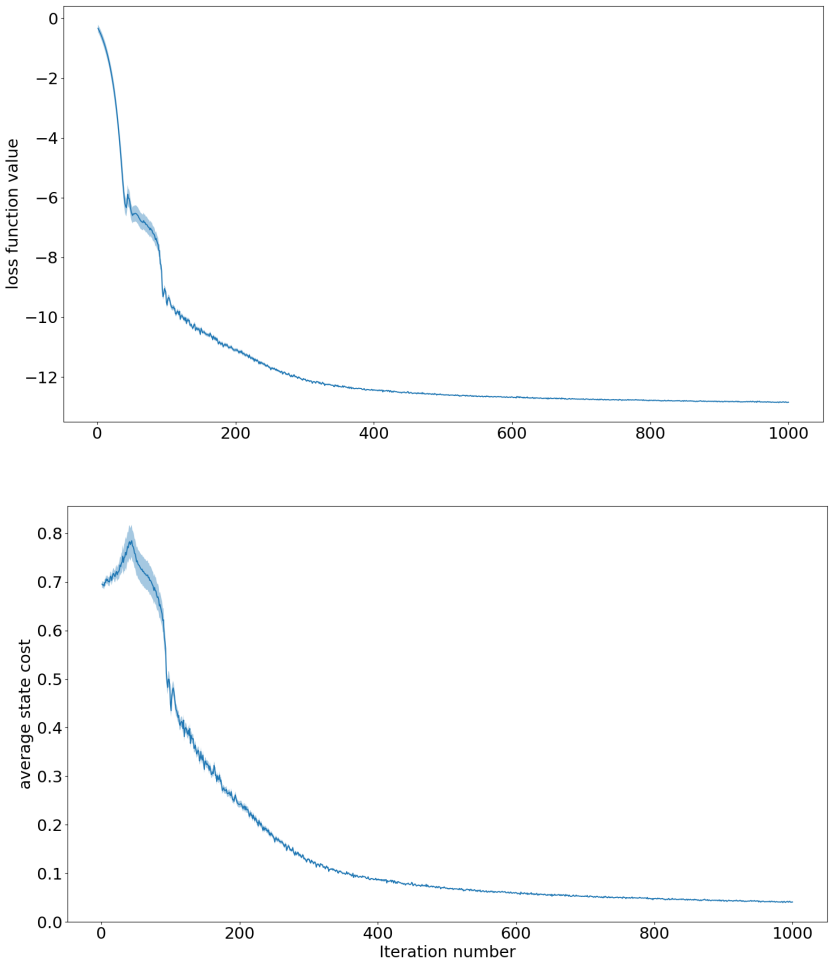
Figure 5.3: Control of the 2D Heat SPDE. (a) shows the desired profile patches and actuator locations for the reaching task. The next three plots show time snapshots from a randomly selected instance of an optimized policy applied to the system. (b) shows the start profile(b), (c) shows half-way through, and (d) shows the end profile. The color-bar depicts the range of temperatures in the simulated field.

We invite the interested reader to refer to Appendix L for specific details on each of our simulations such as cost functions, hyper-parameter values, neural network parameters and videos comparing controlled and uncontrolled SPDEs.

Throughout our simulated experiments, especially for distributed control tasks, we found that the algorithm is not sensitive to the majority of our parameters. We noted that a useful heuristic in applying the algorithm to new problems without having to tune the parameters was to ensure that the starting loss function was not very close to zero (i.e. 1e-10). Despite a large variance of noise that we typically applied to our systems ($\rho = 10$), the optimization algorithm was able to converge in under 1000 iterations for 1D problems and under 2000 iterations for 2D problems.

On the whole, even though injecting higher variance noise into the system inherently makes the control task much more challenging, high variance noise is useful in our algorithm for exploration over rollouts at each iteration. As such, there is an inverse relationship for a given convergence behavior between variance in the noise and number of rollouts.

There are also several interesting behaviors that the IDVRL algorithm demonstrates. First, we noticed that often times throughout optimization, the loss would decrease as desired, but state cost would temporarily increase, before decreasing more dramatically after some number of iterations. This indicates that there may not be a strictly proportional



(b)

Figure 5.4: Convergence of IDVRL Policy for the 1D Heat SPDE. The plots show (a) convergence of the loss function and (b) convergence of the state cost for the IDVRL algorithm over 200 trials of 1000 iterations each for a FNN network.

relationship between loss function and state cost. Indeed a lower state cost implies that the task is being accomplished, yet a trend of decreasing loss function indicated that when there was a temporary increase in state cost, the IDVRL algorithm may have been pushing the network parameters out of a local minimum towards better task performance in later iterations. These trends, depicted in fig. 5.4, indicate that the IDVRL algorithm may perform well on experiments outside the ones considered in this paper.

5.4 Conclusion and Future Directions

This chapter presents a variational reinforcement learning algorithm for the distributed and boundary control of infinite dimensional stochastic systems. The optimization method was derived in Hilbert spaces, thereby avoiding the need to depend on specific descretization schemes to realize the algorithm. The resulting algorithm requires only an actuation model and therefore is mostly model-free. The algorithm was demonstrated on five simulated experiments including 1D and 2D with both distributed and boundary type actuation.

In future work the authors will investigate provable convergence properties for IDVRL based on [113], and will implement the algorithm on some SPDEs described in this paper such as the Stochastic Navier-Stokes equation using state-of-the art CFD solvers. The authors also plan to investigate second-order SPDEs such as the Euler-Bernoulli equation which has been used to investigate the dynamics of tentacle-like soft continuum robots [114].

CHAPTER 6

SPATIO-TEMPORAL STOCHASTIC OPTIMIZATION FOR CONTROL AND CO-DESIGN OF SYSTEMS IN ROBOTICS AND APPLIED PHYSICS

In chapter 5, the IDVRL method is derived for policy optimization and demonstrates efficacy on such problems. However coupled to the problem of optimizing a control policy, is optimizing how the control signal translates to actuation of the system dynamics. This problem is referred to as co-design optimization. Traditionally, one may suggest some actuation design based on some actuation design metric, then optimize a control policy based on the actuation onto the system, and then evaluate the performance on some other actualized system performance metric, thus coupling the optimal control policy to the choices in actuation design.

In this context one wishes a control policy to impart the *most* effect to the system, where it matters, with the *least* control effort. This is often achieved by treating control and co-design separately, applying methods from control theory such as controllability, reachability, or stabilizability, and measuring efficacy of the co-design through linear system gramians. While gramians do not exist for nonlinear systems, perhaps a more critical concern is the decoupling of the co-design optimization problem.

This chapter further develops the approach in chapter 5, and addresses stochastic optimal control and co-design of SPDEs through the lens of stochastic optimization. We propose a joint policy network optimization and actuator co-design optimization strategy via episodic reinforcement that leverages inherent spatio-temporal stochasticity in the dynamics for optimization. The resulting stochastic gradient descent approach bootstraps off the widespread success of SGD methods such as ADAM for training Artificial Neural Networks (ANNs). The stochastic calculus are extended to handle second-order SPDEs in order to address continuum mechanical systems, which in their mathematical treatment resemble

their second-order ODE counterparts prevalent in mechanical systems.

This chapter is motivated by many of the applications of PDEs in robotics, yet primarily seeks to establish capabilities for the eventual design, fabrication, and control of soft-body robots. The behavior of such systems in general follow second order SPDEs. As such, while the proposed methods are general to first and second order systems, we focus our mathematical formulation on second order SPDEs.

In this chapter we tackle the coupled challenge of policy optimization and actuator co-design for SPDEs. Just as in chapters 4 and 5, our approach is founded on the free energy-relative entropy relationship, which is a general principle coming from thermodynamics that also has had success in stochastic optimal control literature [55]

$$\text{Free Energy} \leq \text{Work} - \text{Temperature} \times \text{Entropy} \tag{6.1}$$

We leverage this principle in order to derive a measure-theoretic loss function that utilizes exponential averaging over importance sampled system trajectories in order to choose network and actuator design parameters that simultaneously minimize state cost and control effort. The resulting Spatio-Temporal Stochastic Optimization (STSO) algorithm is applied to a variety of control and co-design problems in fluid mechanics and robotics, culminating in application to a complex nonlinear 2D second-order systems that closely resemble a soft-robotic limb.

6.1 Second Order Soft-Robotic SPDEs in Direct Product Hilbert Spaces

Many complex spatio-temporal systems are given by stochastic partial differential equations of second order in time. Second order SPDEs typically have behavior analogous to second order mechanical SDE systems derived from Newtonian mechanics, in which the actuation acts as external forces or torques which enter through the derivative of the respective linear and rotational momenta. These are typically treated by defining a new momentum state,

and writing the system in matrix-vector form. If one takes a robot arm constrained to one dimension, defined by a scalar second order SDE, and repeatedly adds joints, and thereby degrees of freedom, one would obtain a one-dimensional continuum robot manipulator in the limit, which has infinite degrees of freedom and is described by a suitable one-dimensional second-order SPDE.

6.1.1 The Euler-Bernoulli Continuum System

One such system description is achieved in the simply supported stochastic Euler-Bernoulli equation with Kelvin-Voigt and viscous damping, which is a simplified model of a soft robotic limb. The Euler-Bernoulli equation is used extensively in beam theory, and has applications to a variety of robotic systems beyond soft robotics. Formally, the 1D Euler-Bernoulli equation with Kelvin-Voigt and viscous damping is given in *fields representation* by

$$\begin{aligned}
&\partial_{tt}y + \partial_{xx}\big(\partial_{xx}y + C_d\partial_{xxt}y\big) + \mu\partial_t y = \Phi + \frac{1}{\sqrt{\rho}}\partial_t W(t,x),\\
&y(t,0) = y(t,a) = 0,\\
&y(0,x) = y_0,\\
&\partial_t y(0,x) = v_0,\\
&\partial_{xx}y(t,0) + C_d\partial_{xxt}y(t,0) = 0,\\
&\partial_{xx}y(t,a) + C_d\partial_{xxt}y(t,a) = 0,
\end{aligned} \tag{6.2}$$

where $y(t,x) = y : \mathbb{R}\times\mathscr{D} \to \mathbb{R}$ represents the vertical displacement of the beam over problem domain $\mathscr{D}$, C_d represents the Kelvin-Voigt damping coefficient, μ represents the viscous damping coefficient, and all functional dependencies of the nonlinear policy Φ have been dropped since it has a different form in the fields representation. Note that the policy and stochastic effects enter as forces. With the change of variables $v := \partial_t y$, this system has the

typical second order matrix-vector form

$$\partial_t \begin{bmatrix} y \\ v \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -A_0 & -C_d A_0 - \mu \end{bmatrix} \begin{bmatrix} y \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \Phi + \begin{bmatrix} 0 \\ \frac{1}{\sqrt{\rho}} \end{bmatrix} \partial_t W(t,x), \tag{6.3}$$

where $A_0 = \partial_{xxxx}$ without boundary conditions. Now, we lift this SPDE into infinite dimensional Hilbert spaces. Define $Y \in \mathscr{H}$ as the Hilbert space analog of $y(t,x)$, $V \in \mathscr{H}$ as the Hilbert space analog of $v(t,x)$, and a variable $Z := Y \times V$ on the direct product Hilbert space $\mathscr{H}^2 := \mathscr{H} \times \mathscr{H}$. Note that $Z \in \mathscr{H}^2$ is a Hilbert space analog of a variable $z(t,x) = [y(t,x) \;\; v(t,x)]^\top \in \mathbb{R}^2$. In Hilbert spaces, A_0 becomes an operator acting on $\mathscr{H}$ and 1 gets replaced by the identity operator I acting on $\mathscr{H}$. Rewriting eq. (6.3) in Hilbert space semi-linear form yields

$$\mathrm{d}Z = \mathscr{A} Z \mathrm{d}t + G\Big(\Phi(t, Z, \mathbf{x}; \Theta^{(k)})\mathrm{d}t + \frac{1}{\sqrt{\rho}} \mathrm{d}W(t)\Big), \tag{6.4}$$

where $\mathscr{A} : \mathscr{H}^2 \to \mathscr{H}^2$ is the linear operator $\mathscr{A} = [0 \;\; I; -A_0 \;\; -C_d A_0 - \mu I]$, $G : \mathscr{H} \to \mathscr{H}^2$ is an operator representing how control and spatio-temporal noise enter the system $G = [0; I]$, and $\mathrm{d}W(t)$ is a Cylindrical Wiener process on $\mathscr{H}$. Note that the Hilbert space variables Y, V, and Z no longer have spatial dependence as the Hilbert space vectors capture the spatial continuum over which the problem is defined.

6.1.2 Detailed Models of Soft-Robotic Limbs

While the Euler-Bernoulli SPDE has wide applicability, it relies on a small-angle assumption, which is not suitable for some soft-robotic applications such as in soft-robotic manipulators or end-effectors. In [12] the errors introduced by violation of this approximation have been reduced significantly by parametrizing the beam's backbone by a tangent angle and including a single-parameter hysteretic term. The resulting model is used for a hyper-flexible system.

However, the majority of modern soft-body robotics modeling research typically deviates from the Euler-Bernoulli approach. The majority of modern models of soft-robotic systems are divided into two main categories: constant curvature approximation, and non-constant curvature approximation [115]. For a detailed review of constant and piecewise constant curvature methods as of 2010, refer to [116]. More recent constant curvature methods include [117], wherein the authors use the principle of virtual power to derive a model with constant curvature segments and discrete torsional joints. In [118], the authors start with a piecewise constant curvature approximation, and produce a model that they then validate against a piecewise constant curvature robotic manipulator. In [119], the authors apply a piecewise constant curvature model to a continuum robotic manipulator actuated by pressure differentials provided by bellows. Constant and piecewise constant curvature models are often much simpler in implementation, yet these models can fail when the system does not have a uniform shape or is acted on by a large external load. The interested reader can refer to [32] for a recent review of control methodologies on constant and piecewise constant curvature models of continuum manipulators.

Non-constant curvature models are increasing in popularity due to a typically more accurate representation of a continuum system. They are typically broken into three subcategories: continuum approximations of hyperredundant models, spring-mass models, and cosserat or geometrically exact models. For a complete review of design, fabrication and control strategies that sweep across the discipline of soft robotics, the interested reader should refer to [120].

The continuum approximation of hyperredundant models were among the first proposed continuum methods [121], and led to several interesting applications [122, 123, 124]. On the other end of the spectrum, Cosserat models are currently the most exact models of continuum systems. Derived from the context of beam theory, these geometrically exact models often have a large number of PDE states, making them quite difficult to simulate at high frequency. Yet, their high fidelity has made them an appealing research direction.

Cosserat models have been simulated in real-time for very slender rods with uniform cross-section in [125, 126]. In [115], the authors develop a geometrically exact 3-dimensional (3D) model on Lie groups of a tentacle-like tapered soft robot arm actuated by cables, resulting in a PDE with 18 states. Similar models are also developed and validated in [127]. In addition to introducing significant model complexity, Cosserat PDEs are also known to suffer from *stiff* dynamics with respect to the Courant-Friedrichs-Lewy stability condition [126].

In between these two extremes are the set of continuum spring-mass models. These models often emerge in studies of biological systems, such as the appendages of the octopus vulgaris [128, 129, 130]. Their features include accurate, volume preserving representations of muscular forces and lower PDE state dimensionality compared to Cosserat models. In [131], the authors derive a particle-based model that falls into the category of spring-mass models, and they establish a link between such particle systems and continuum mechanics. In this chapter we consider a stochastic variant of their model, actuated by an actuation function modeled after muscular behavior common to cephalopods [130].

The SPDE governing the dynamics of a continuum elastic material is formally given by

$$\rho_m \partial_{tt} \mathbf{s} = \operatorname{div}(\sigma) + \mathbf{f}_g + \Phi + \frac{1}{\sqrt{\rho}} \partial_t \mathbf{W}, \tag{6.5}$$

where $\rho_m \in \mathbb{R}$ is the material density, $\mathbf{s} = \mathbf{s}(t,x,y)$ is the deformed state, $\sigma = \sigma(t,x,y)$ is the stress tensor, $\mathbf{f}_g$ is the force of gravity, Φ is the vector nonlinear policy, and $\mathbf{W} = \mathbf{W}(t,x,y)$ is a vector Cylindrical Wiener noise process. The material state $\mathbf{s}$ can be expressed as the sum of an initial rest state $\mathbf{r}$ and deformation $\mathbf{d}$, each of which are parameterized over 2D domain $\mathscr{D} = X \times Y$.

$$\mathbf{s}(t,x,y) = \mathbf{r}(x,y) + \mathbf{d}(t,x,y). \tag{6.6}$$

The total stress tensor σ in eq. (6.5) is the sum of the elastic and viscous stresses,

$$\sigma(t,x,y) = \sigma^{\varepsilon}(t,x,y) + \sigma^{\nu}(t,x,y). \tag{6.7}$$

Assuming linear elasticity, the elastic stress tensor σ^{ε} may be related to the strain tensor $\varepsilon = \varepsilon(t,x,y)$ via the the stiffness tensor $\mathbf{C}$ as

$$(\sigma^{\varepsilon})_{ij} = (\mathbf{C})_{ijkl}(\varepsilon)_{kl}, \tag{6.8}$$

where subscript of a parenthesis $(A)_{ijkl}$ denotes tensor element i,j,k,l of A, and Einstein summation notation is utilized to perform tensor contractions. Strains, denoted ε, within the material are determined by Green's strain tensor, where subscripts of s indicate partial derivatives with respect to coordinates x or y, as

$$\varepsilon = \begin{bmatrix} \|\mathbf{s}_x\|^2 - 1 & \frac{1}{2}\langle \mathbf{s}_x, \mathbf{s}_y \rangle \\ \frac{1}{2}\langle \mathbf{s}_x, \mathbf{s}_y \rangle & \|\mathbf{s}_x\|^2 - 1 \end{bmatrix}. \tag{6.9}$$

For isotropic materials, entries of the stiffness tensor $\mathbf{C}$ are determined by tensile constant $\zeta \in \mathbb{R}$ and shear modulus $\mu \in \mathbb{R}$

$$(\mathbf{C})_{iiii} = \zeta, \quad (\mathbf{C})_{ijij} = (\mathbf{C})_{jiij} = \frac{1}{2}\mu, \quad i \neq j. \tag{6.10}$$

Dissipative effects can be modelled by Kelvin-Voigt damping, which adds a viscous stress $\sigma^{\nu} = \sigma^{\nu}(t,x,y)$ proportional to the strain rate $\nu = \nu(t,x,y)$,

$$(\nu)_{ij} = \frac{\mathrm{d}(\varepsilon)_{ij}}{\mathrm{d}t}, \tag{6.11}$$

$$(\sigma^{\nu})_{ij} = (\mathbf{D})_{ijkl}(\nu)_{kl}. \tag{6.12}$$

The damping tensor $\mathbf{D}$ is proportional to the stiffness tensor $\mathbf{C}$ by a retardation time constant

$\tau \in \mathbb{R}$

$$\mathbf{D} = \tau \mathbf{C}. \tag{6.13}$$

In this case, the resulting SPDE can be again lifted into Hilbert spaces in a similar fashion as in eq. (6.2). Define displacement velocity $\mathbf{u} := \partial_t \mathbf{s} = \partial_t \mathbf{d}$, and rewrite eq. (6.5) in typical second order matrix-vector form

$$\partial_t \begin{bmatrix} \mathbf{d} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{d} \\ \mathbf{u} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{\mathrm{div}(\boldsymbol{\sigma}) + \mathbf{f}_g}{\rho_m} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{\rho_m} \end{bmatrix} \Phi + \begin{bmatrix} 0 \\ \frac{1}{\rho_m \sqrt{\rho}} \end{bmatrix} \partial_t \mathbf{W}. \tag{6.14}$$

We again lift this SPDE into infinite dimensional Hilbert spaces. Note that $\mathbf{d} = \mathbf{d}(t,x,y)$ and $\mathbf{u} = \mathbf{u}(t,x,y)$ have x and y components that are each defined over 2D spatial domain $\mathscr{D} = X \times Y$. Define $\mathscr{W} \in \mathscr{H} \times \mathscr{H}$ as the Hilbert space analog of $\mathbf{d}(t,x,y)$, $\mathscr{V} \in \mathscr{H} \times \mathscr{H}$ the Hilbert space analog of $\mathbf{u}(t,x,y)$, and a variable $Z := \mathscr{W} \times \mathscr{V}$ on the direct product Hilbert space $\mathscr{H}^4 := \mathscr{H} \times \mathscr{H} \times \mathscr{H} \times \mathscr{H}$. This new variable $Z \in \mathscr{H}^4$ is a Hilbert space analog of a variable $\mathbf{z}(t,x,y) := [\mathbf{d}(t,x,y) \;\; \mathbf{u}(t,x,y)]^\top \in \mathbb{R}^4$. Rewriting eq. (6.14) in Hilbert space semi-linear form yields

$$\mathrm{d}Z = \mathscr{A} Z \mathrm{d}t + F(Z)\mathrm{d}t + G\Big(\Phi(t,Z,\mathbf{x};\Theta^{(k)})\mathrm{d}t + \frac{1}{\sqrt{\rho}}\mathrm{d}W(t)\Big), \tag{6.15}$$

where $\mathscr{A} : \mathscr{H}^4 \to \mathscr{H}^4$ is a linear operator, F is the nonlinear operator which contains the forces due to stresses and gravity, $G : \mathscr{H}^2 \to \mathscr{H}^4$ is an operator representing how the $\mathscr{H}^2$-valued control policy and spatio-temporal noise enter the system, and $\mathrm{d}W(t)$ is a Cylindrical Wiener process on $\mathscr{H}^2$. Again, the Hilbert space variables lose spatial dependence as they represent the entire spatial continuum.

6.2 Girsanov Theorem for Second Order SPDEs

The proposed approach is derived from the perspective of a measure theoretic view of variational optimization, wherein the change of measures, or RN derivative, is a tool that is widely leveraged to change the sampling distribution of an expectation. Thus, such a framework requires a properly formulated Girsanov theorem for second order SPDEs defined on time-indexed Hilbert spaces.

Theorem 6.1 (Girsanov)**.** *Let Ω be a sample space with a σ-algebra $\mathscr{F}$. Consider the following $\mathscr{H}^2$-valued (or similarly $\mathscr{H}^4$-valued) nonlinear stochastic processes*

$$\mathrm{d}Z = \big(\mathscr{A}Z + F(t,Z)\big)\mathrm{d}t + \frac{1}{\sqrt{\rho}}G(t,Z)\mathrm{d}W(t), \tag{6.16}$$

$$\mathrm{d}\tilde{Z} = \big(\mathscr{A}\tilde{Z} + F(t,\tilde{Z})\big)\mathrm{d}t + G(t,\tilde{Z})\left(\tilde{B}(t,\tilde{Z})\mathrm{d}t + \frac{1}{\sqrt{\rho}}\mathrm{d}W(t)\right), \tag{6.17}$$

where $\tilde{B}$ is a nonlinear functional mapping into $\mathscr{H}$ (or similarly $\mathscr{H}^2$), $Z(0) = \tilde{Z}(0) = z_0$ and $W \in \mathscr{H}$ (or similarly $W \in \mathscr{H}^2$) is a Cylindrical Wiener process with respect to measure $\mathbb{P}$. Assume eqs. (6.16) *and* (6.17) *are well posed and have unique weak $\mathscr{F}_t$-adapted solutions $Z(t)$ and $\tilde{Z}(t)$, $t \geq 0$. Let Γ be a set of continuous-time, infinite-dimensional trajectories in the time interval $[0,T]$. Define the probability law of Z over trajectories Γ as $\mathscr{L}(\Gamma) := \mathbb{P}(\omega \in \Omega | Z(\cdot,\omega) \in \Gamma)$. Similarly, define the law of $\tilde{Z}$ as $\tilde{\mathscr{L}}(\Gamma) := \mathbb{P}(\omega \in \Omega | \tilde{Z}(\cdot,\omega) \in \Gamma)$. Assume*

$$\mathbb{E}_{\mathbb{P}}\big[e^{\frac{1}{2}\int_0^T ||\psi(s)||^2 \mathrm{d}s}\big] < +\infty, \tag{6.18}$$

where

$$\psi(t) := \sqrt{\rho}\tilde{B}\big(t, Z(t)\big) \in \mathscr{H}. \tag{6.19}$$

Then

$$\tilde{\mathscr{L}}(\Gamma) = \mathbb{E}_{\mathbb{P}}\left[\exp\left(\int_0^T \big\langle \psi(s), \mathrm{d}W(s)\big\rangle - \frac{1}{2}\int_0^T \big|\big|\psi(s)\big|\big|^2 \mathrm{d}s\right)\middle| Z(\cdot) \in \Gamma\right]. \tag{6.20}$$

Proof. Define the process

$$\hat{W}(t) := W(t) - \int_0^t \psi(s)\mathrm{d}s. \tag{6.21}$$

Under the above assumption, $\hat{W}$ is a Cylindrical Wiener process with respect to a measure $\mathbb{Q}$ defined by

$$\mathrm{d}\mathbb{Q}(\omega) = \exp\left(\int_0^T \left\langle \psi(s), \mathrm{d}W(s)\right\rangle - \frac{1}{2}\int_0^T \left|\left|\psi(s)\right|\right|^2 \mathrm{d}s\right)\mathrm{d}\mathbb{P} \tag{6.22}$$

$$= \exp\left(\int_0^T \left\langle \psi(s), \mathrm{d}\hat{W}(s)\right\rangle + \frac{1}{2}\int_0^T \left|\left|\psi(s)\right|\right|^2 \mathrm{d}s\right)\mathrm{d}\mathbb{P}. \tag{6.23}$$

The proof that $\hat{W}$ is a Cylindrical Wiener process with respect to $\mathbb{Q}$ can be found in [2, Theorem 10.14]. Now, using eq. (6.21), eq. (6.16) is rewritten as

$$\mathrm{d}Z = \big(\mathscr{A}Z + F(t,Z)\big)\mathrm{d}t + \frac{1}{\sqrt{\rho}}G(t,Z)\mathrm{d}W(t) \tag{6.24}$$

$$= \big(\mathscr{A}Z + F(t,Z)\big)\mathrm{d}t + G(t,Z)\left(B(t,Z)\mathrm{d}t + \frac{1}{\sqrt{\rho}}\mathrm{d}\hat{W}(t)\right). \tag{6.25}$$

Notice that the SPDE in eq. (6.25) has the same form as eq. (6.17). Therefore, under the introduced measure $\mathbb{Q}$ and noise profile $\hat{W}$, $Z(\cdot,\omega)$ becomes equivalent to $\tilde{Z}(\cdot,\omega)$. Conversely, under measure $\mathbb{P}$, eq. (6.24) (or eq. (6.25)) behaves as the original system in eq. (6.16). In other words, eq. (6.16) and eq. (6.25) describe the same system on $(\Omega, \mathscr{F}, \mathbb{P})$. From the uniqueness of solutions and the aforementioned reasoning, one has

$$\mathbb{P}\big(\{\tilde{Z} \in \Gamma\}\big) = \mathbb{Q}\big(\{Z \in \Gamma\}\big).$$

The result follows from eq. (6.23). □ □

The notion most pertinent to the subsequent derivation is the change of measures, or RN derivative, between the associated measures of the uncontrolled and controlled

systems defined in eq. (6.16) and eq. (6.17), respectively. First, note that for any function $\lambda \in C([0,T];H)$ one has [2, Chapter 1]

$$\mathbb{E}_{\mathbb{P}}[\lambda(Z)] = \int_{\Omega} \lambda\big(Z(\cdot,\omega)\big)\mathrm{d}\mathbb{P}(\omega) = \int_{C([0,T];H)} \lambda(\mathsf{x})\mathrm{d}\mathscr{L}(\mathsf{x}), \quad \mathsf{x} \in \Gamma. \tag{6.26}$$

Thus, from eq. (6.20), one can obtain

$$\mathrm{d}\tilde{\mathscr{L}} = \exp\bigg(\int_0^T \big\langle \psi(s), \mathrm{d}\hat{W}(s) \big\rangle + \frac{1}{2}\int_0^T \big|\big|\psi(s)\big|\big|^2 \mathrm{d}s \bigg) \mathrm{d}\mathscr{L}, \tag{6.27}$$

which directly leads to the RN derivative

$$\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}} = \exp\bigg(-\int_0^T \big\langle \psi(s), \mathrm{d}\hat{W}(s) \big\rangle - \frac{1}{2}\int_0^T ||\psi(s)||^2 \mathrm{d}s \bigg). \tag{6.28}$$

In the case of semilinear SPDEs of the form eq. (6.4) and similarly any semilinear SPDE in the general form eq. (2.13), the function ψ which defines this RN derivative is given by

$$\psi(t) := \sqrt{\rho}\Phi(t, Z, \mathbf{x}; \Theta^{(k)}), \tag{6.29}$$

which simplifies the RN derivative in eq. (6.28) to

$$\begin{aligned} \frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}} = \exp\bigg(&-\sqrt{\rho}\int_0^T \big\langle \Phi(s, Z, \mathbf{x}; \Theta^{(k)}), \mathrm{d}\hat{W}(s) \big\rangle \\ &- \frac{\rho}{2}\int_0^T \big|\big|\Phi(s, Z, \mathbf{x}; \Theta^{(k)})\big|\big|^2 \mathrm{d}s \bigg). \end{aligned} \tag{6.30}$$

For convenience, we assign functions to each term in eq. (6.30)

$$\mathscr{N}(\Theta, \mathbf{x}) := \int_0^T \big\langle \Phi(s, Z, \mathbf{x}; \Theta^{(k)}), \mathrm{d}\hat{W}(s) \big\rangle, \tag{6.31}$$

$$\mathscr{P}(\Theta, \mathbf{x}) := \int_0^T \big|\big|\Phi(s, Z, \mathbf{x}; \Theta^{(k)})\big|\big|^2 \mathrm{d}s. \tag{6.32}$$

6.3 Spatio-Temporal Stochastic Optimization

The proposed measure theoretic framework was first derived in chapter 5 for the simpler case of policy optimization without co-design optimization. In this chapter, it is extended to policy and actuator co-design optimization. These frameworks are explicit feedback formulation of the feedforward and MPC formulations given in chapter 4. The explicit feedback is realized through the nonlinear policy $\Phi(t, Z, \mathbf{x}; \Theta^{(k)})$, which is a potentially time-varying policy that has explicit state dependence.

Nonlinear, explicit state dependence allows for a feedback policy that can extract pertinent information from the state for control, and is in a sense reactive to undesired evolutions of the state. Policy networks have had widespread success in extracting pertinent features in a multitude of systems, and are utilized here for the nonlinear policy. Embedded in this function is also a dependence on $\mathbf{x}$, which describes how the actuator may depend on some design variables, such as actuator placement in the spatial domain. This approach also encompasses cases where terms that parametrize how the actuators are shaped or sized are included in the nonlinear policy.

The proposed framework is based on an instantiation of the second law of thermodynamics given in eq. (6.1) in the following form [107, 57]

$$-\frac{1}{\rho}\log\mathbb{E}_{\mathscr{L}}\left[\exp(-\rho J)\right] = \min_{\Theta,\mathbf{x}}\left[\mathbb{E}_{\tilde{\mathscr{L}}}(J) + \frac{1}{\rho}D_{KL}(\tilde{\mathscr{L}}\,||\mathscr{L})\right], \tag{6.33}$$

where $J = J(X)$ is an arbitrary state cost functional and the notation $\mathbb{E}_{\tilde{\mathscr{L}}}(\cdot)$ is an expectation with respect to the path measure $\tilde{\mathscr{L}}$, i.e. it is a path integral expectation. Relating eq. (6.33) to eq. (6.1), the metaphorical work and entropy describe a metaphorical energy landscape for which there is a minimizing measure. Sampling from this measure would simultaneously minimize state cost and the KL-divergence term, which is interpreted as control effort. The

measure that optimizes eq. (6.33) is the so-called Gibbs measure

$$\mathrm{d}\mathscr{L}^* = \frac{\exp(-\rho J)\mathrm{d}\mathscr{L}}{\mathbb{E}_{\mathscr{L}}\big[\exp(-\rho J)\big]}. \tag{6.34}$$

The significance of eq. (6.33) from the perspective of optimal control theory lies in established connections between eq. (6.33) and the HJB equation in infinite dimensions, as shown in [57]. This connection to a foundational principle in optimal control literature motivates the use of eq. (6.33) and the resulting optimal measure in eq. (6.34) for the derivation of the proposed measure-theoretic optimization strategy.

It is not known how to sample directly from the Gibbs measure in eq. (6.34). Instead, variational optimization methods are often used to iteratively minimize the controlled distribution's "distance" [1] to the Gibbs measure [108, 57, 132]. Define the control policy and actuator co-design problem as

$$\Theta^* := \underset{\Theta}{\operatorname{argmin}} D_{KL}(\mathscr{L}^* || \tilde{\mathscr{L}}) \tag{6.35a}$$

$$\mathbf{x}^* := \underset{\mathbf{x}}{\operatorname{argmin}} D_{KL}(\mathscr{L}^* || \tilde{\mathscr{L}}). \tag{6.35b}$$

Typically, control and co-design problems are formulated separately, and in such a context, actuator co-design optimization with policy optimization must be performed in alternating outer and inner loops, respectively. However, this chapter develops a joint optimization problem which develops a path integral graph of trajectory rollouts, and penalizes actuator design and policy design on a common metric. This is made possible through our measure theoretic approach, which yields a path integral loss function. In this context, it is prudent to apply joint optimization as opposed to alternating optimization. A concise understanding of joint optimization of the resulting path integral graph is described in section 6.7. To make joint optimization clear, define a new variable $\hat{\Theta} := [\Theta, \mathbf{x}]^\top$, and

[1] Distance here is defined in the KL sense, and is abusive terminology since the KL-Divergence is non-symmetric, and therefore not a distance metric in the mathematical sense.

with it the new joint variational optimization as

$$\hat{\Theta}^* = \underset{\hat{\Theta}}{\operatorname{argmin}} D_{KL}(\mathscr{L}^* || \tilde{\mathscr{L}}). \tag{6.36}$$

Expanding the KL divergence and applying the chain rule yields

$$\hat{\Theta}^* = \underset{\hat{\Theta}}{\operatorname{argmin}} \left[\int \log \Big(\frac{\mathrm{d}\mathscr{L}^*}{\mathrm{d}\mathscr{L}} \frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}} \Big) \mathrm{d}\mathscr{L}^* \right], \tag{6.37}$$

which is equivalent to minimizing

$$\hat{\Theta}^* = \underset{\hat{\Theta}}{\operatorname{argmin}} \left[\int \log \Big(\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}} \Big) \mathrm{d}\mathscr{L}^* \right]. \tag{6.38}$$

Performing importance sampling yields

$$\hat{\Theta}^* = \underset{\hat{\Theta}}{\operatorname{argmin}} \left[\int \log \Big(\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}} \Big) \frac{\mathrm{d}\mathscr{L}^*}{\mathrm{d}\mathscr{L}} \frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}} \mathrm{d}\tilde{\mathscr{L}} \right]. \tag{6.39}$$

The proposed iterative approach performs episodic reinforcement with respect to a loss function in order to optimize eq. (6.39). Define the loss function as

$$L(\hat{\Theta}) := \mathbb{E}_{\tilde{\mathscr{L}}} \left[\log \Big(\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}} \Big) \frac{\mathrm{d}\mathscr{L}^*}{\mathrm{d}\mathscr{L}} \frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}} \right]. \tag{6.40}$$

Plugging eq. (6.28) and eq. (6.34) into eq. (6.40) yields

$$L(\hat{\Theta}^{(k)}) = \mathbb{E}_{\tilde{\mathscr{L}}} \left[\frac{\exp(-\rho \tilde{J})}{\mathbb{E}_{\tilde{\mathscr{L}}} \big[\exp(-\rho \tilde{J}) \big]} \Big(- \sqrt{\rho} \mathscr{N}(\hat{\Theta}^{(k)}) - \frac{\rho}{2} \mathscr{P}(\hat{\Theta}^{(k)}) \Big) \right], \tag{6.41}$$

where $\tilde{J} = \tilde{J}(Z_{0:T}, \hat{\Theta}^{(k)})$ is defined as

$$\tilde{J}(Z_{0:T}, \hat{\Theta}^{(k)}) := J(Z_{0:T}) + \frac{1}{\sqrt{\rho}} \mathscr{N}(\hat{\Theta}^{(k)}) + \frac{1}{2} \mathscr{P}(\hat{\Theta}^{(k)}), \tag{6.42}$$

and $J(Z_{0:T})$ is a state cost evaluated over the state trajectory $Z_{0:T}$. For reaching tasks, $J(Z_{0:T})$ is typically a weighted 2-norm distance to the goal state.

This loss function is understood in the path integral sense, i.e. it is an importance-sampled path integral expectation over a Gibbs-averaged performance metric. In other words, it compares sampled trajectory rollouts by evaluating them on the exponentiated $\tilde{J}$ performance metric. The importance sampling terms $\mathscr{N}$ and $\mathscr{P}$, which appear in L and $\tilde{J}$ add a quadratic control penalization term and a mixed control noise term. In the Loss function, they serve as weights for the exponentiated cost trajectories. For convenience, we denote the exponentiated cost term as

$$\mathscr{E}(Z_{0:T}, \Theta^{(k)}) := \frac{\exp\big(-\rho \tilde{J}(Z_{0:T}, \Theta^{(k)})\big)}{\mathbb{E}_{\tilde{\mathscr{Z}}}\Big[\exp\big(-\rho \tilde{J}(Z_{0:T}, \Theta^{(k)})\big)\Big]}. \tag{6.43}$$

Recall, that the nonlinear policy Φ is a functional mapping into Hilbert space $\mathscr{H}$ (or $\mathscr{H}^2$). This is kept general for derivation purposes, however it implies that the nonlinear policy controls each element of an infinite vector in Hilbert space $\mathscr{H}$ (or $\mathscr{H}^2$). A more realistic, but less general representation refines the policy as

$$\Phi(t, Z, \mathbf{x}; \Theta^{(k)}) = \mathbf{m}(\mathbf{x})^\top \varphi(Z; \Theta^{(k)}), \tag{6.44}$$

where $\mathbf{m}(\mathbf{x}) : \mathscr{D}^N \to \mathbb{R}^N \times \mathscr{H}$ represents the effect of the actuation from N actuators on the infinite-dimensional field. Typically this is either a Gaussian-like exponential with mean centered at the actuator locations or an indicator function.

In eq. (6.44), $\varphi(Z; \Theta^{(k)}) : \mathscr{H} \to \mathbb{R}^N$ is a policy network with N control outputs representing N distributed (or boundary) actuators. Note that as desired, the tensor contraction given on the right hand side of eq. (6.44) produces a vector in $\mathscr{H}$ (or $\mathscr{H}^2$). Splitting the actuation function from the control signal is also desired because we ultimately wish to use a finite input, finite output policy network for the function $\varphi(Z; \Theta^{(k)})$. The inner product

terms become

$$\mathscr{N}(\hat{\Theta}^{(k)}) = \int_0^T \left\langle \mathbf{m}(\mathbf{x})^\top \varphi(Z;\Theta^{(k)}), \mathrm{d}\hat{W}(s) \right\rangle, \tag{6.45}$$

$$\begin{aligned}\mathscr{P}(\hat{\Theta}^{(k)}) &= \int_0^T ||\mathbf{m}(\mathbf{x})^\top \varphi(Z;\Theta^{(k)})||^2 \mathrm{d}s \\ &= \int_0^T \left\langle \varphi(Z;\Theta^{(k)}), \mathbf{M}(\mathbf{x})\varphi(Z;\Theta^{(k)}) \right\rangle \mathrm{d}s,\end{aligned} \tag{6.46}$$

where $\mathbf{M}(\mathbf{x}) := \mathbf{m}(\mathbf{x})\mathbf{m}(\mathbf{x})^\top \in \mathbb{R}^{N\times N}$.

Many state of the art methods for training networks rely on a gradient approach [109, 110], wherein one computes a loss function that depends on the network parameters, and iteratively updates the network parameters based on the gradients of the loss with respect to said network parameters. Bootstrapping off the wide-spread success of these methods, we prescribe a similar gradient-descent update, which can be interchanged with any such gradient approach, given by

$$\Theta^{(k+1)} = \Theta^{(k)} - \gamma_\Theta \nabla_\Theta L(\Theta^{(k)}, \mathbf{x}^{(k)}), \tag{6.47}$$

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \gamma_\mathbf{x} \nabla_\mathbf{x} L(\Theta^{(k)}, \mathbf{x}^{(k)}), \tag{6.48}$$

where γ_Θ and $\gamma_\mathbf{x}$ are the learning rates for the policy parameters and actuator design parameters, respectively, and $\nabla_\mathbf{a} := \frac{\partial}{\partial \mathbf{a}}$, denotes the partial derivative with respect to some finite-dimensional vector a.

Figure 6.1 is a graphical representation of our approach. A Hilbert space policy network with initialized weights is passed through an SPDE model or physical realization of the system to produce state trajectories, which are used to compute a state cost as well as a sparse tensor that is used to compute the inner products in a memory and time-efficient manner. This method will be explained further in the subsequent section. These terms are used together to compute, by Monte-Carlo approximation, the path integral expectation in the loss function.

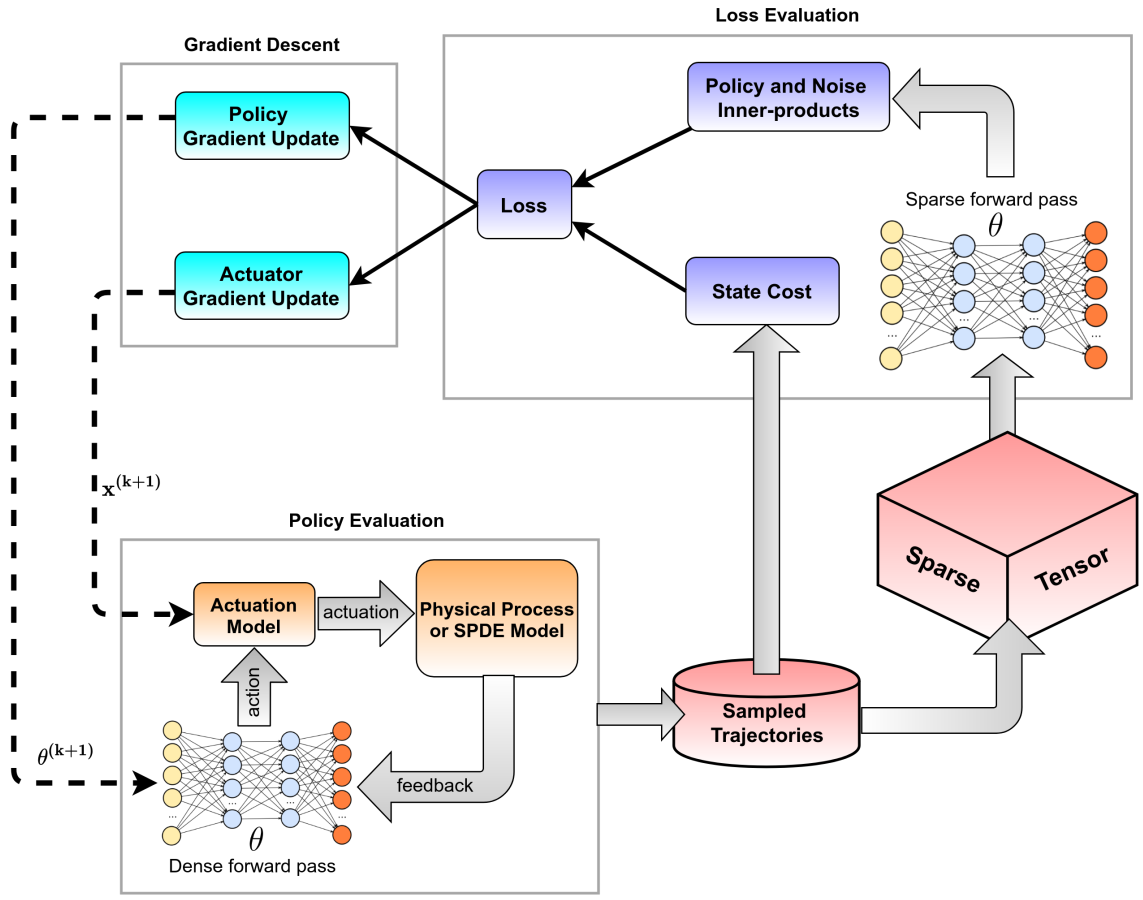


Figure 6.1: Diagram of the spatio-temporal stochastic optimization (STSO) approach for policy and actuator co-design optimization.

These state trajectory rollouts, together with the associated performance metric, importance sampling terms, Gibbs-averaged term, and resulting loss function can be thought of as forming a connected graph between the loss function and the optimization variables. Gradients are computed across the connections in this graph to produce gradients of the loss with respect to the policy variables and with respect to the actuator design variables. These gradients follow numerous gradient paths and are used in conjunction with a gradient-based optimization algorithm such as SGD to provide parameter updates to the policy network and actuator design. This approach is independent of discretization scheme, choice of actuation design components, choice of state cost functional, and choice of policy network.

A key observation is that up to this point, we have a continuous-time optimization

approach defined completely in Hilbert spaces; we have not performed any discretization of time or space. The benefit of this fact is that it equips our framework with the property of being *discretization agnostic*. In other words, *any* discretization scheme, for *any* semi-linear SPDE with additive Cylindrical Wiener process can be used in conjunction with the proposed algorithm. In fact, since the only term from the dynamics to appear in eq. (6.40) and eq. (6.42) is the Cylindrical Wiener process $\mathrm{d}W$, the optimization approach may consider the system and actuation model as a differentiable *black-box*; one needs only the model of the additive Cylindrical Wiener process. In what follows, we consider *any* discretization of the system, and provide numerical methods to handle difficulties that arise after discretization.

6.4 Discrete Approximation Methods

The above derivation provides a general Hilbert space framework for optimizing nonlinear policies to control SPDEs to achieve some task. This approach represents an *optimize-then-discretize* scheme. In order to implement the approach as an algorithm on a digital computer, data must be collected at finite frequency from interactions with a real system, or generated by a discretized physics-based or data-based model. In this section, we address the implement-ability of our approach with these considerations in mind.

6.4.1 Sparse Spatial Integration

Unique to this approach for training policy networks are the inner products that appear in eq. (6.45) and eq. (6.46). Each of these Hilbert space inner products represent a spatial integration over the finite region $\mathscr{D}$. Numerical methods to efficiently compute these spatial inner products were first developed in chapter 5. Consider the inner product in eq. (6.46).

For 2D systems, it can be represented as a spatial integral in the form

$$\int_0^T \left\langle \varphi(Z(s);\Theta^{(k)}), \mathbf{M}(\mathbf{x})\varphi(Z(s);\Theta^{(k)}) \right\rangle \mathrm{d}s \tag{6.49}$$

$$= \int_0^T \iint_D \varphi(Z(s,x,y);\Theta^{(k)})^\top \mathbf{M}(x,y)\varphi(Z(s,x,y);\Theta^{(k)})\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}s$$

$$= \int_0^T \sum_{j=1}^{\infty} \varphi(Z(s,e_j);\Theta^{(k)})^\top \mathbf{M}(e_j)\varphi(Z(s,e_j);\Theta^{(k)})\mathrm{d}s, \tag{6.50}$$

where $\{e_j \in \mathscr{H} : j = 0,1,2,\dots\}$ forms an orthonormal basis over $\mathscr{H}$. After applying a spatial discretization to the system, the basis becomes a finite set $\{e_j \in \mathbb{R}^{J^2} : j = 0,1,2,\dots\}$, where J is the number of discretization points in each dimension [2]. One choice of such a basis is the set of one-hot vectors which emerges naturally from applying a central difference discretization, however, one may use a different basis or project onto the one-hot basis. Therefore, this integration scheme is also agnostic to the choice of discretization. Thus, evaluating the spatial integral is reduced to summing up forward passes through the policy network with each pixel considered individually.

Motivated by this one-hot basis approach, in chapter 5 we developed a sparse matrix method for efficiently handling the spatial integrals, which become integrals of time-indexed (J^2, J, J) tensors for each sample. The key observation is that since the basis elements of each (J,J) "image" have only one activated "pixel", the resulting tensor is tremendously sparse. As such, each layer's activation can be computed with a sparse matrix multiplication, resulting in the so-called *SparseForwardPass* method that is not memory intensive for relatively large 2D problems. This can be applied to policy network architectures that utilize fully connected layers and convolutional layers. For convolutional input layers, this can be achieved by representing the convolution as a matrix multiplication with a Toeplitz-like matrix constructed from the filter coefficients [111].

[2] We assume without loss of generality, that each dimension has the name number of discretization points J.

6.4.2 Approximate Discrete Optimization

On the side of actuator co-design optimization, it is useful to refine the optimization procedure in eqs. (6.47) and (6.48) for certain optimization variables in light of the discretization. Such a case would be the placement of actuators, where depending on the actuation function, the system may not feel the effect of an actuator placed between discretization points.

To see this more clearly, consider the 1D spatial continuum $\mathscr{D} = [0,1]$ discretized into a 11 point 1D grid. Lets assume that an actuator is chosen to be placed at $x = 0.25$. Even though the actuation function $\mathbf{m}(\mathbf{x})$ may be Gaussian-like function, the majority of the actuation will be felt in between two grid points, namely 0.2 and 0.3. This problem is even more severe if the actuation function $\mathbf{m}(\mathbf{x})$ is the indicator function, as there will be *no* actuation exerted on the field *irrespective* of the control signal magnitude. Denote the number of spatial discretization points as J and a 3D discretized problem domain grid as $\hat{\mathscr{D}}$ composed of J^3 elements. Let $\mathbf{x}_p$ denote the subset of optimization variables of $\mathbf{x}$ that capture the placement of actuators, and $\mathbf{x}_c$ as the rest of the elements of $\mathbf{x}$. That is, $\mathbf{x} = [\mathbf{x}_c \;\; \mathbf{x}_p]^\top$. The optimization problem becomes

$$\begin{aligned} &\min_{\boldsymbol{\Theta},\mathbf{x}} \; L(\boldsymbol{\Theta},\mathbf{x}) \\ &\text{subject to} \;\; \mathbf{x}_p \in \hat{\mathscr{D}}. \end{aligned} \tag{6.51}$$

This formulation is an accurate representation, yet limits gradient flow from the loss function back to the actuator design parameters. In order to maintain these gradients, [103] develops the following approximate approach. Define a one-to-one map $S : \hat{\mathscr{D}} \to \mathbb{Z}_+$, where $\mathbb{Z}_+$ denotes the set of positive integers. Applying the forward and inverse mapping produces

a gradient-based parameter update of the form

$$\Theta^{(k+1)} = \Theta^{(k)} - \gamma_{\Theta} \nabla_{\Theta} L(\Theta^{(k)}, \mathbf{x}_p^{(k)}, \mathbf{x}_c^{(k)}), \tag{6.52}$$

$$\mathbf{x}_c^{(k+1)} = \mathbf{x}_c^{(k)} - \gamma_{\mathbf{x}_c} \nabla_{\mathbf{x}_c} L(\Theta^{(k)}, \mathbf{x}_p^{(k)}, \mathbf{x}_c^{(k)}), \tag{6.53}$$

$$\mathbf{x}_p^{(k+1)} = S^{-1}\left(R\Big(S\big(\mathbf{x}_p^{(k)} - \gamma_{\mathbf{x}_p} \nabla_{\mathbf{x}_p} L(\Theta^{(k)}, \mathbf{x}_p^{(k)}, \mathbf{x}_c^{(k)})\big)\Big)\right), \tag{6.54}$$

where $R(\cdot)$ simply rounds to the nearest integer, and we have used the overloaded notation $L(\Theta^{(k)}, \mathbf{x}_p^{(k)}, \mathbf{x}_c^{(k)}) = L(\Theta^{(k)}, \mathbf{x}^{(k)})$. We will use this overloaded for this and other functions for simplicity. This approach allows the use of well-known gradient update algorithms such as ADA-Grad [109] and ADAM [110]. See [133] for an overview of popular gradient update algorithms used in machine learning.

6.4.3 Modified Virtual Approximate Discrete Optimization

There is a key limitation with the above approach. In the case of a small gradient, the effect of rounding may "override" the effect of the gradient update. Thus, the gradient update may be prevented from changing the value until the gradient is large enough to push the variable close to the next discretization point. This effect becomes especially pronounced in the case of a coarse discretization, but also becomes apparent when the actuator placement is near an optimal value. In this local region, the gradient is relatively flat, so improper tuning of the learn rate combined with a coarse discretization grid would result in convergence to a sub-optimal value.

Here we propose the following novel modification. Consider a virtual optimization variable $\mathbf{v} \in \mathscr{D}$ to serve as an intermediary in the optimization process. The goal of this intermediary is to preserve the gradient movement from the update, yet only allow the true optimization variable $\mathbf{x}_p \in \hat{\mathscr{D}}$ to exist on the discretization grid. Instead of applying the $S^{-1}\Big(R\big(S(\cdot)\big)\Big)$ map to the same variable update as in eq. (6.54), we wish to carry the true gradient update information over iterations, so that the effect of the iterative update

is additive over iterations. However, the issue is that the map $S^{-1}\Big(R\big(S(\cdot)\big)\Big)$ is a non-differentiable map due to the rounding in $R(\cdot)$. The proposed solution is to modify the optimization problem as follows

$$\Theta^{(k+1)} = \Theta^{(k)} - \gamma_{\Theta}\nabla_{\Theta}L(\Theta^{(k)}, \mathbf{x}^{(k)}), \tag{6.55}$$

$$\mathbf{x}_c^{(k+1)} = \mathbf{x}_c^{(k)} - \gamma_{\mathbf{x}_c}\nabla_{\mathbf{x}_c}L(\Theta^{(k)}, \mathbf{x}_p^{(k)}, \mathbf{x}_c^{(k)}), \tag{6.56}$$

$$\mathbf{v}^{(k+1)} = \mathbf{v}^{(k)} - \gamma_{\mathbf{x}_p}\nabla_{\mathbf{x}_p}L(\Theta^{(k)}, \mathbf{x}_p^{(k)}, \mathbf{x}_c^{(k)}), \tag{6.57}$$

$$\mathbf{x}_p^{(k+1)} = S^{-1}\Big(R\big(S(\mathbf{v}^{(k+1)})\big)\Big). \tag{6.58}$$

Here, we carry two variables: a continuous-valued variable $\mathbf{v} \in \mathscr{D}$, and a discrete-valued variable $\mathbf{x}_p \in \hat{\mathscr{D}}$. We compute the gradient of the loss L with respect to the *discrete*-valued variable $\nabla_{\mathbf{x}_p}L(\Theta^{(k)}, \mathbf{x}_p^{(k)}, \mathbf{x}_c^{(k)})$, but apply this gradient in a gradient update to the *continuous*-valued variable $\mathbf{v}$, which in effect bypasses the non-differentiable map. This important difference results in a virtual optimization variable $\mathbf{v}$ which can reach the true optimal value, but is not applied to the system, and a secondary variable $\mathbf{x}_p$ which represents the grid element nearest to the optimal value, and is applied to the system.

6.5 Algorithm and Network Architecture

As discussed previously, implementation of the above framework requires spatial and temporal discretization of the SPDEs discussed in section 2.2. With this in mind, we choose an ANN for our nonlinear policy $\varphi(Z;\Theta^{(k)})$. In this chapter we use FNNs for 1D experiments, and CNNs for 2D experiments. We use physics-based models of each SPDE to generate training data. Given that the proposed framework is semi-model-free, real system data can seamlessly replace the physics-based model as described in chapter 5. We only need prior knowledge of the flavor of noise, a differentiable model[3] of the actuation function $\mathbf{m}(\mathbf{x})$, and the actuator design elements $\mathbf{x}$.

[3]Note that the actuation model can also be a black-box model

Algorithm 5 Spatio-Temporal Stochastic Optimization

Function: $\Theta^* = \textit{OptimizePolicyActuatorVars}(T, K, R, Z_0, N, \rho, \Delta t, \mu, \sigma_\mu^{(0)}, \Theta^{(0)}, \mathbf{x}_p^{(0)}, \gamma_\Theta, \gamma_{\mathbf{x}_c}, \gamma_{\mathbf{x}_p})$
for $k = 0$ to K **do**
  Compute $\mathbf{m}(\mathbf{x}_p^{(k)}, \mathbf{x}_c^{(k)}), \mathbf{M}(\mathbf{x}_p^{(k)}, \mathbf{x}_c^{(k)})$
  for $r = 0$ to R **do**
    for $t = 0$ to T **do**
      $\mathrm{d}W_t^r \leftarrow SampleNoise()$
      $Z_t^r \leftarrow Propagate(Z_{t-1}^r, \Theta^{(k)}, \mathrm{d}W_t^r)$ via eq. (6.4)
      $\Phi_t^r \leftarrow SparseForwardPass(\Theta^{(k)}, Z_t^r)$
    end for
  end for
  $J^{0:R} \leftarrow StateCost(Z_{0:T}^{0:R})$
  $N^{0:R} \leftarrow \mathscr{N}\left(\Phi_{0:T}^{0:R}, \mathrm{d}\hat{W}_{0:T}^{0:R}, \mathbf{m}(\mathbf{x}^{(k)})\right)$ via eq. (6.45)
  $P^{0:R} \leftarrow \mathscr{P}\left(\Phi_{0:T}^{0:R}, \mathbf{M}(\mathbf{x}^{(k)})\right)$ via eq. (6.46)
  $E^{0:R} \leftarrow \mathscr{E}(J^{0:R}, N^{0:R}, P^{0:R})$ as in eq. (6.43)
  $L \leftarrow ComputeLoss(P^{0:R}, N^{0:R}, E^{0:R})$ via eq. (6.41)
  Compute $\nabla_\Theta L$ via backprop
  Compute $\nabla_{\mathbf{x}_p} L$ via backprop
  Compute $\nabla_{\mathbf{x}_c} L$ via backprop
  $\Theta^{(k+1)} \leftarrow GradientStep(\nabla_\Theta L, \gamma_\Theta, \Theta^{(k)})$ via eq. (6.55)
  $\mathbf{x}_c^{(k+1)} \leftarrow GradientStep(\nabla_{\mathbf{x}_c} L, \gamma_{\mathbf{x}_c}, \mathbf{x}_c^{(k)})$ via eq. (6.56)
  $\mathbf{v}^{(k+1)} \leftarrow GradientStep(\nabla_{\mathbf{x}_p} L, \gamma_{\mathbf{x}_p}, \mathbf{v}^{(k)})$ via eq. (6.57)
  $\mathbf{x}_p^{(k+1)} \leftarrow SnapToGrid(\mathbf{v}^{k+1})$ via eq. (6.58)
end for

The resulting modified algorithm, modified from the original algorithm named Actuator Design and Policy Learning (ADPL) in [103], is referred to here as STSO and shown in algorithm 5. Here we modify the notation as well to specify the role of rollouts by using superscript r to denote rollout $r \in \{r_i\}_{i=0}^R$, superscript $0:R$ to denote the collected set of rollouts, subscript t to denote time instant t of a variable, subscript $0:T$ to denote the collected set of a variable along a trajectory, and have again used the overloaded notation $\mathbf{m}(\mathbf{x}_p, \mathbf{x}_c) = \mathbf{m}(\mathbf{x})$ and $\mathbf{M}(\mathbf{x}_p, \mathbf{x}_c) = \mathbf{M}(\mathbf{x})$. We also generalize to optimizing over actuator placement and other non-placement actuator design variables, such as actuator variance. The inputs can change depending on the specific problem but in most cases contain time horizon (T), number of iterations (K), number of rollouts (R), initial state (Z_0), number of

actuators (N), noise variance (ρ), time discretization (Δt), initial actuator variance ($\sigma_\mu^{(0)}$), initial network parameters ($\Theta^{(0)}$), initial actuator locations ($\mathbf{x}_p^{(0)}$), policy learn rate (γ_Θ), actuator location learn rate ($\gamma_{\mathbf{x}_p}$), and actuator shape learn rate ($\gamma_{\mathbf{x}_c}$). For more information on *SampleNoise*(), refer to [95, Chapter 10].

The method *GradientStep* performs a gradient update of any gradient-based optimization algorithm. We apply ADAM [110] gradient update variants of eqs. (6.55) to (6.57) in all of the experiments in this chapter. This version of *GradientStep* is different than that of [103] due to the modifications described in section 6.4.3, namely there is no need to add a separate [4] momentum term to help the gradients reach optimal values since we are now carrying the continuous-valued virtual variable $\mathbf{v}$, which can change over iterations even when the true variable $\mathbf{x}_p$ remains at the previous grid element due to the method *SnapToGrid*.

The use of different learning rates for each type of variable is often essential. The authors conjecture that the optimization landscape is typically more shallow for the actuator design than for the policy parameters. For most of the experiments, the actuator placement learning rate $\gamma_{\mathbf{x}_p}$ is set to about 30 times larger than the policy network learning rate γ_Θ, however this can be dependent on the problem, selection of number of actuators, and policy parameter initialization type (e.g. Xavier vs zeroes).

6.6 Policy & Co-Design Optimization of Simulated Robotics PDEs

In [103] the approach was applied to four simulated SPDE experiments to simultaneously place actuators and optimize a policy network. Each experiment used less than 32 GB RAM, and was run on a desktop computer with a Intel Xeon 12-core CPU with a NVIDIA GeForce GTX 980 GPU. The code was written to operate inside a Tensorflow graph [112] to leverage rapid static graph computation, as well as sparse linear algebra operations used by *SparseForwardPass* [63]. The first two experiments involved a reaching task, where the

[4] separate from the momentum native to the ADAM update

SPDEs are initialized at a zero initial condition over the spatial region and must reach certain values at pre-specified regions of the spatial domain. The last two experiments involved a suppression task, where some non-zero initial condition must be suppressed on desired regions.

The data that was used for training was generated by a spatial central difference, semi-implicit time discretized version of each SPDE. These schemes are described in detail in [95, Chapter 3 & 10]. Each experiment had all actuator locations initialized by sampling from a uniform distribution on $[0.4a, 0.6a]$, where a denotes the spatial size. For 3500 iterations of the algorithm, run times for the most complicated system–the Euler-Bernoulli equation–were about 15 hours. Details of the experiments and videos of the controlled systems can be found in the provided links [5]. We encourage the interested reader to contact the first author for code.

Each of the experiments in this section utilized FNNs for the nonlinear policy $\varphi(h;\Theta)$, built in a Tensorflow graph [112] with two hidden layers of Rectified Linear Unit (ReLU) neurons. All policy network weights were initialized with the Xavier initialization [134] and trained with ADAM [110]. In every experiment the function $\mathbf{m}(\mathbf{x})$ was modeled as a Gaussian-like exponential function with the means co-located with the actuator locations, and considered a state cost functional of the form

$$J := \sum_t \sum_x \kappa \big(h_{\text{actual}}(t,x) - h_{\text{desired}}(t,x)\big)^2 \cdot \mathbb{1}_S(x), \tag{6.59}$$

where κ is a state cost weight, and $\mathbb{1}_S(x)$ is defined by

$$\mathbb{1}_S(x) := \begin{cases} 1, & \text{if } x \in S \\ 0, & \text{otherwise,} \end{cases} \tag{6.60}$$

where S is the spatial subregion on which the desired profile is defined. These desired spatial

[5] Supplement: tinyurl.com/yc7fq3lc | Video: https://youtu.be/yo48a6JqKE0

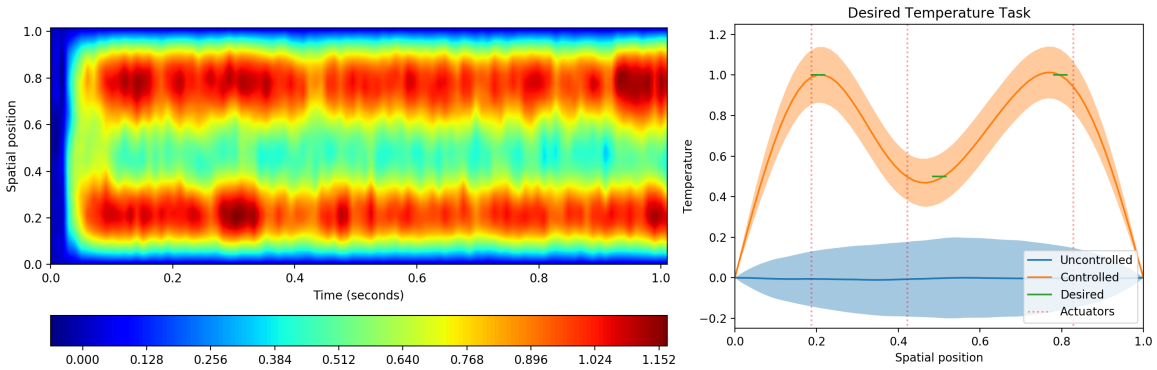


Figure 6.2: Heat Equation Temperature Reaching Task. (left) controlled contour plot of a randomly selected trajectory rollout where color represents temperature, (right) final time snapshot comparing to the uncontrolled system. Mean trajectories are represented with a solid line, while a 2σ standard deviation is represented with a shaded region.

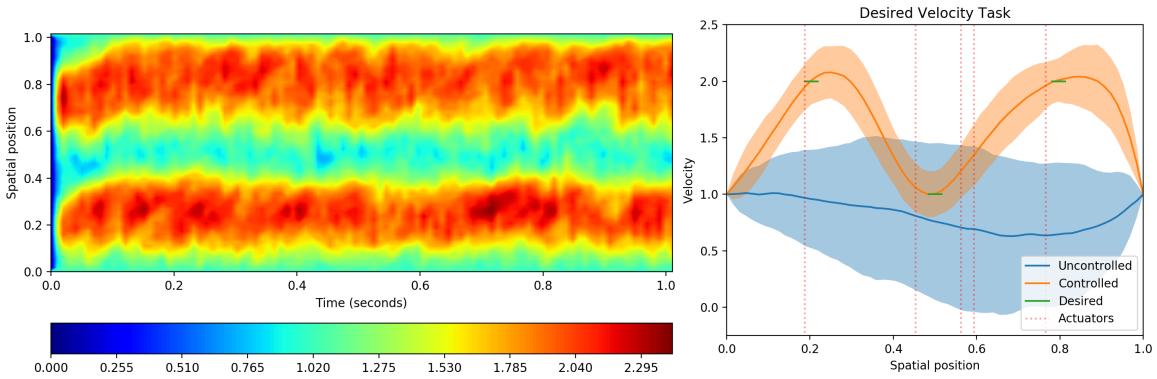


Figure 6.3: Burgers Velocity Reaching Task. (left) controlled contour plot of a randomly selected trajectory rollout where color represents velocity, (right) final time snapshot comparing to the uncontrolled system. Mean trajectories are represented with a solid line, while a 2σ standard deviation is represented with a shaded region.

regions were different for each experiment, and are depicted as green bars in the associated figures.

The first experiment was a temperature reaching task on the 1D Heat equation with homogeneous Dirichlet boundary conditions, given in *fields representation* by

$$\begin{aligned} \partial_t h(t,x) &= \varepsilon \partial_{xx} h(t,x) + \mathbf{m}(\mathbf{x})^\top \varphi\big(h(t,x);\Theta\big) + \frac{1}{\sqrt{\rho}} \partial_t W(t,x), \\ h(t,0) &= h(t,a) = 0, \\ h(0,x) &= h_0(x), \end{aligned} \tag{6.61}$$

where ε is the thermal diffusivity parameter. The results of 3000 iterations of optimization with 200 trajectory rollouts per iteration are depicted in fig. 6.2. The task was to raise the temperature at regions specified in green to specified values depicted in the figure.

The next experiment was a velocity reaching task on the Burgers equation with non-homogenous Dirichlet boundary conditions, given in *fields representation* by

$$\begin{aligned} \partial_t h(t,x) &= -h(t,x)\partial_x h(t,x) + \varepsilon \partial_{xx} h(t,x) + \mathbf{m}(\mathbf{x})^\top \boldsymbol{\varphi}\big(h(t,x);\Theta\big) \\ &\quad + \frac{1}{\sqrt{\rho}} \partial_t W(t), \\ h(t,0) &= h(t,a) = 1.0, \\ h(0,x) &= h_0(x), \end{aligned} \tag{6.62}$$

where the parameter ε is the viscosity of the medium. The results of 3500 iterations with 100 trajectory rollouts per iteration are depicted in fig. 6.3. The Burgers equation is often used as a simplified model of fluid flow, however Burgers-like reaction-advection-diffusion PDE are also often used to describe swarms of robotic systems [10]. The Burgers equation has a nonlinear advection term, which produces an apparent rightward motion. The algorithm appears to have taken advantage of the advection for actuator placement in order to solve the task with lower control effort.

The heat equation is a pure diffusion SPDE, while the Burgers equation shares the diffusion term with the Heat equation with an added advection term. The results of the Heat and Burgers experiments show actuator locations that take advantage of the natural behavior of each SPDE. In the case of the Heat equation, actuators are near the desired regions such that the temperature profile can reach a flat peak of the diffusion at the desired profile. In the case of the Burgers equation, the advection pushes towards the right end of the space, thus forming a wave front that develops at the right end, but leaves the left end dominated by the diffusion term. This is again reflected in the placement of actuators. The first actuator is near the desired region just as the actuators in the Heat SPDE, while two of the actuators between

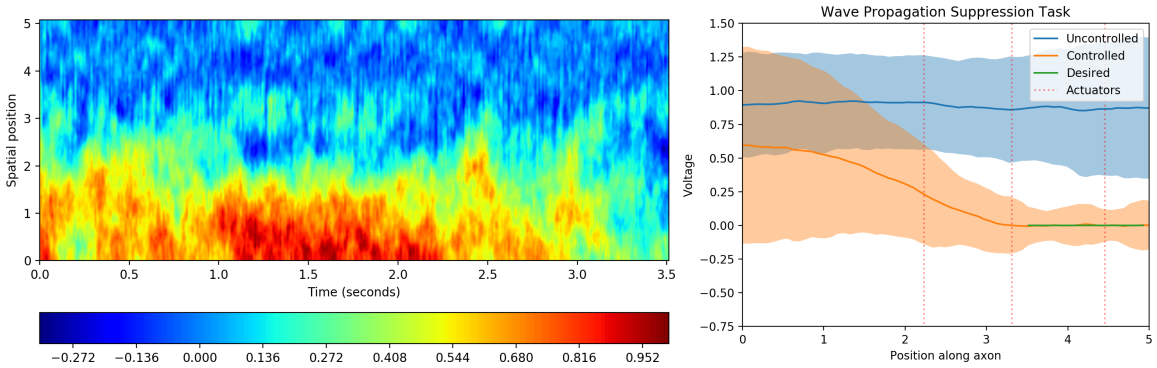


Figure 6.4: Nagumo Suppression Task. (left) controlled contour plot of a randomly selected trajectory rollout where color represents voltage, (right) final time snapshot comparing to the uncontrolled system. Mean trajectories are represented with a solid line, while a 2σ standard deviation is represented with a shaded region.

the center and the right region are located to be able to control the amplitude and shape of the developing wave front so as to produce a flat peak that aligns with the desired region at the desired velocity. The central desired region is flanked on both sides by actuators that are nearly equidistant, in order to produce another desired flat velocity region at this location.

The third experiment was a voltage suppression task on the Nagumo equation with homogeneous Neumann boundary conditions, given in *fields representation* by

$$\begin{aligned} \partial_t h(t,x) &= \varepsilon \partial_{xx} h(t,x) + h(t,x)\big(1-h(t,x)\big)\big(h(t,x)-\alpha\big) \\ &\quad + \mathbf{m}(\mathbf{x})^\top \varphi\big(h(t,x);\Theta\big) + \frac{1}{\sqrt{\rho}}\partial_t W(t,x), \\ h_x(t,0) &= h_x(t,a) = 0, \\ h(0,x) &= \left(1+\exp\left(-\frac{2-x}{\sqrt{2}}\right)\right)^{-1}, \end{aligned} \tag{6.63}$$

where the parameter $\alpha = -0.5$ determines the speed of a wave traveling down the length of the extent and $\varepsilon = 1.0$ determines the rate of diffusion. The Nagumo equation is often used in neuroscience as a model of the propagation of voltage across an axon in neuronal activation [95]. However, it has also been used in robotics applications, such as in [11], where it was used to describe robot navigation in crowded environments.

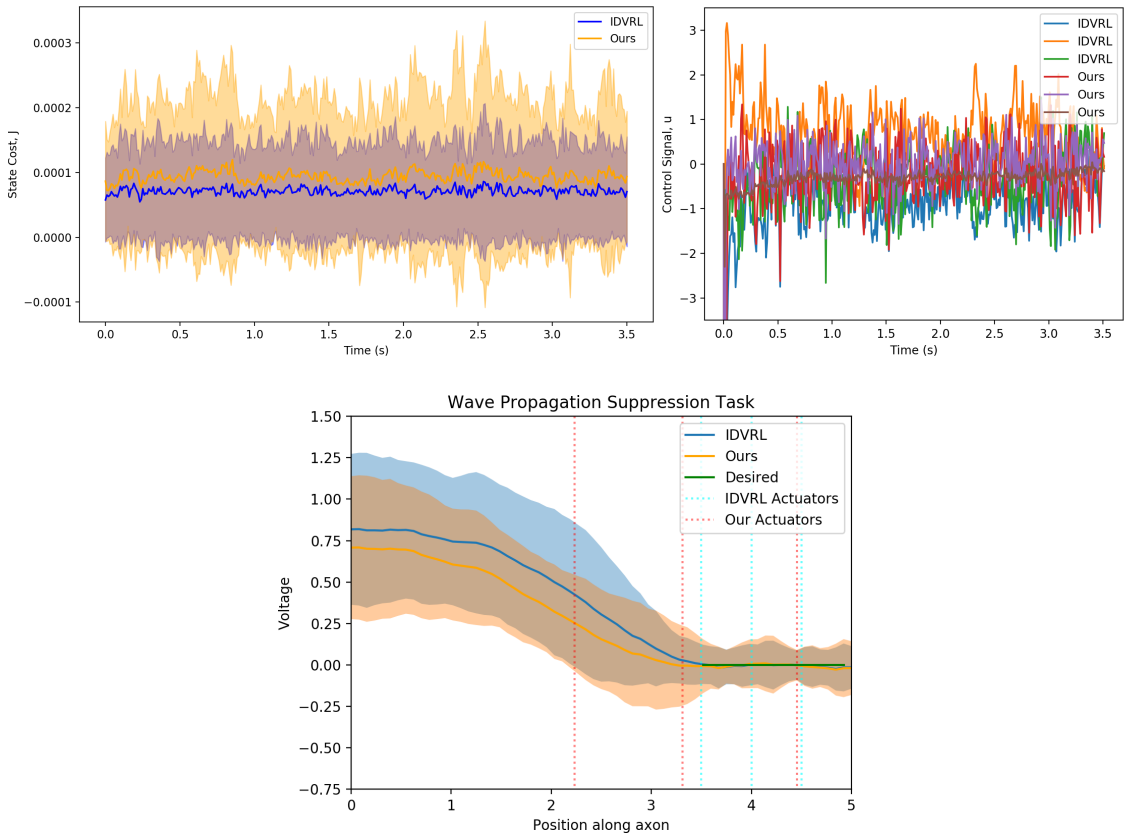


Figure 6.5: Nagumo Suppression Task Comparison Plots. (top left) controlled state cost plot, where solid lines denote mean, and shaded regions denote a 2σ standard deviation, (top right) Control signal comparison plot, where lines represent mean behavior, and (bottom) Final time snapshot comparing the actuators placed by our approach and actuators placed by a human expert with policy optimization by IDVRL.

The joint policy and co-design optimization results are depicted in fig. 6.4. The task was to suppress an initial voltage on the left end, that without intervention propagates toward the right end, as shown by the uncontrolled trajectories. The Nagumo equation in eq. (6.63) is composed of a diffusive term and a 3rd-order nonlinearity, making this equation the most challenging 1D SPDE from a nonlinear control perspective. Despite this, our approach was able to simultaneously place actuators and provide control such that the task was solved. The algorithm was run for 2000 iterations, and demonstrates actuator placement optimization that takes advantage of the natural system behavior. This task was also the most challenging due to the significantly longer planning horizon of 3.5 seconds, as compared to the 1.0 second planning horizon of all the other experiments.

In order to validate our proposed approach, we compared the actuator locations that the algorithm found after optimization to the actuator locations that were hand placed by a human expert for the simulated experiments conducted in our prior work [63]. To have a valid comparison, we ran the IDVRL algorithm for both sets of actuator locations. Figure 6.5 reports these results. The left figure shows that the state costs for each are almost identical. Note that the scale here is 10^{-4}. The center figure shows the control signals for each actuator, for each method, and demonstrates that for almost identical state cost values, the control effort for each actuator with our approach is lower on average. The calculated average control signal magnitudes for human-placed actuators are 3.3 times higher than those placed by our method. The third plot shows the voltage profile at the final time. We hypothesize that the lower control effort is due to the control over the shape of the spatially propagating signal, enabling it to have a smoother transition into the desired region. While the penalty of this actuator placement is a slightly higher variance on the desired region, the choice appears correct given the result.

The final task conducted in [103] was an oscillation suppression task on the Euler-Bernoulli equation with Kelvin-Voigt damping given in eq. (6.2), and is depicted in fig. 6.6. As shown, the initial condition prescribes spatial oscillations, that then oscillate temporally. The second-order nature of the system creates offset and opposite oscillations in the velocity profile, that in turn produce offset and opposite oscillations in the position profile. Without interference, the oscillations proceed over the entire time window, as shown in the left subfigure of fig. 6.6. As shown in the right subfigure of fig. 6.6, our approach successfully suppresses these oscillations, which die out quickly under the given control policy. In this experiment, the actuators remained inside the initialized actuator placement region $[0.4a, 0.6a]$ prescribed for all experiments.

The Euler-Bernoulli oscillation suppression task is in fact very challenging and complex. Producing a control signal at an actuator location that is in phase with the velocity oscillations will amplify the oscillations, leading to a divergence. The actuator location and control

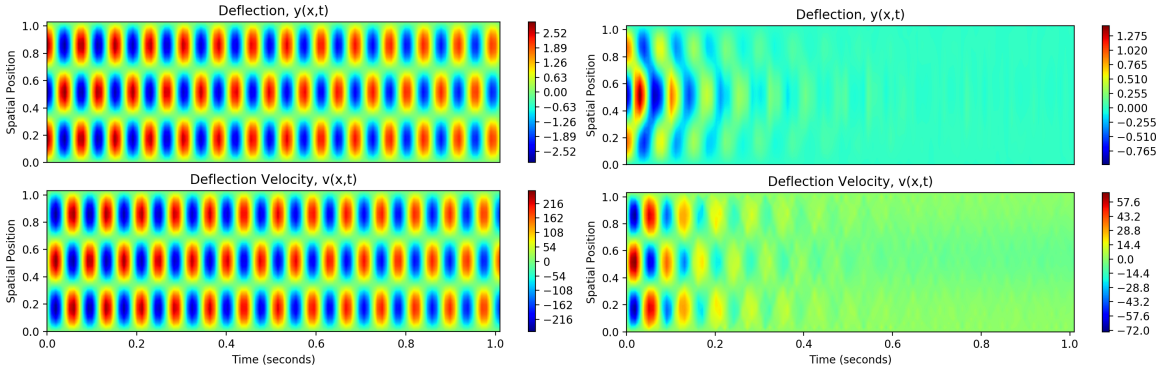


Figure 6.6: Euler-Bernoulli Suppression Task. (left) Uncontrolled contour plot (right) Controlled contour plot. In both plots, color represents deflection on top, and deflection velocity on bottom.

signal from the policy network must work in concert to produce a control signal out of phase with the velocity and matching its frequency, which is time varying due to control. This time-varying frequency is depicted in the left subfigure of fig. 6.6.

Each of the above 1D experiments have unique challenges and in most cases the spatio-temporal problem space produces a joint policy optimization and actuator co-design problem that is littered with local minima. These experiments demonstrate that the proposed approach can jointly optimize a policy network and actuator design. These results and the overall performance of the algorithm are indicative that this approach may enable actuator design on problem spaces where a human has little to no prior knowledge to rely on in a-priori designing actuation to solve the problem by hand.

6.6.1 Scaling to Higher Dimensions

The above experiments motivated the novel experiments presented here, which scale policy and actuator co-design optimization to large 2D problem spaces. The primary challenges with scaling to higher dimensions are related to the memory storage requirements of large computational graphs. As discussed in the previous section, each of the prior experiments were performed on a static TensorFlow graph, which in many cases has an advantage in runtime performance, yet requires significant memory pre-allocation. Scaling policy and

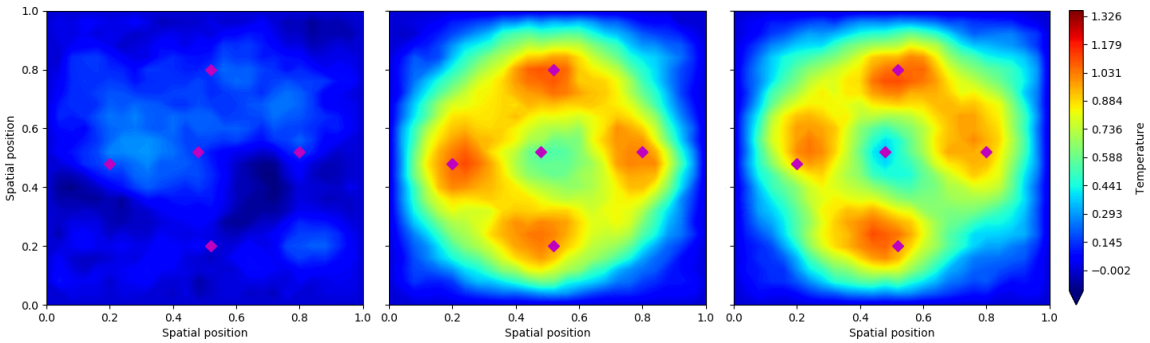


Figure 6.7: Controlled 2D Heat Equation Contours of a random trajectory rollout with actuators denoted in magenta and color spectrum denoting temperature. (left) initial time contour with spatially random initial condition (center) half-way time contour (right) final time contour.

actuator co-design optimization to 2D spaces often requires more memory storage than 64 GB in a static graph setting. As such, it is recommended to balance dynamic allocation with static graph computation.

For this task, the goal is to control the 2D Heat equation with homogeneous Dirichlet boundary conditions. The goal is to raise the temperature in four outer regions to $T = 1.0$ and raise the temperature in one central region to $T = 0.5$. The results of 5000 iterations, with 100 rollouts per iteration are depicted in fig. 6.7. Similar to the 1D heat equation, here the pure diffusive nature of the Heat equation is best leveraged by placing actuators as close to the desired regions as possible, as is demonstrated by the algorithm in this case. A video of the controlled state evolution is available at the link provided [6].

The SPDE is spatially discretized using a central difference discretization into 25 grid points on each axis for a total of 625 states, and is temporally discretized with a semi-implicit time discretization. This problem has a dramatic increase in scale compared to the 1D examples given above, which typically had 64 states.

In this case five actuators were provided to the system to achieve this task, and were all intitialized by sampling x and y locations from a uniform distribution over $[0, a]$, where a is the side length. The nonlinear policy network for this experiment utilized two convolutional

[6] Video: https://youtu.be/yo48a6JqKE0

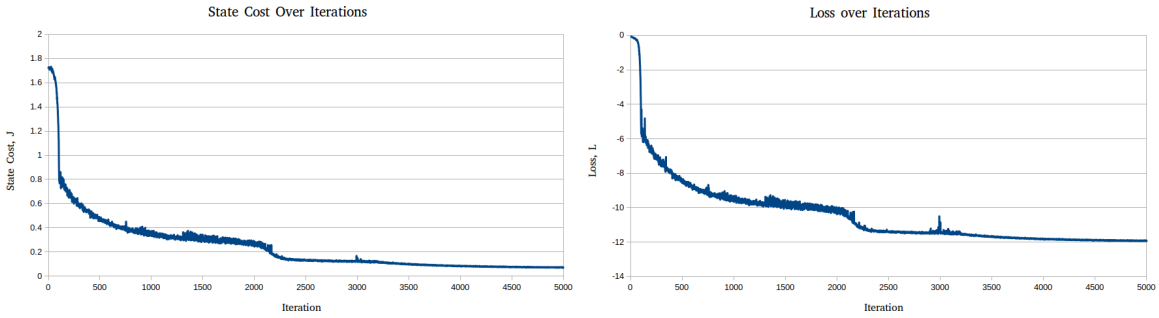


Figure 6.8: Convergence of State Cost and Loss for 2D Heat Equation. (left) state cost over iterations for 5000 iterations (right) loss over iterations for 5000 iterations.

layers, two max-pool layers, and a fully connected output layer, with ReLU activations throughout. The policy network weights were initialized with the Xavier initialization [134].

The convergence behavior of the algorithm over 5000 iterations for the obtained solution is depicted in fig. 6.8. As depicted, there is a close correlation between the behavior of the state cost and the behavior of the loss, which is desirable in many cases. However, in many cases this approach can violate strict proportionality between the loss and the state cost in the near term in order to obtain dramatic state cost improvements in the long term. This is reported in greater detail in our prior work [63].

6.6.2 Policy & Co-Design Optimization of a Soft Robotic Limb

In the final experiment, the algorithm was applied to a 2D soft manipulator governed by the dynamics in eq. (6.5). These dynamics exhibit complex nonlinear behavior that in many ways present the most challenging task that has been conducted with our approach; the 2D PDE dynamics are nonlinear, 2nd-order, and stochastic.

For this experiment the task was to jointly optimize the policy and placement of actuators such that the soft limb deflected by one unit vertically while maintaining initial tip position in x, subject to a highly exaggerated gravitational force two orders of magnitude larger than nominal. The exaggerated gravitational force models an external force preventing task completion and forces better actuation design performance for task completion. The result of 4000 iterations of the algorithm with 50 rollouts per iteration are depicted in fig. 6.9.

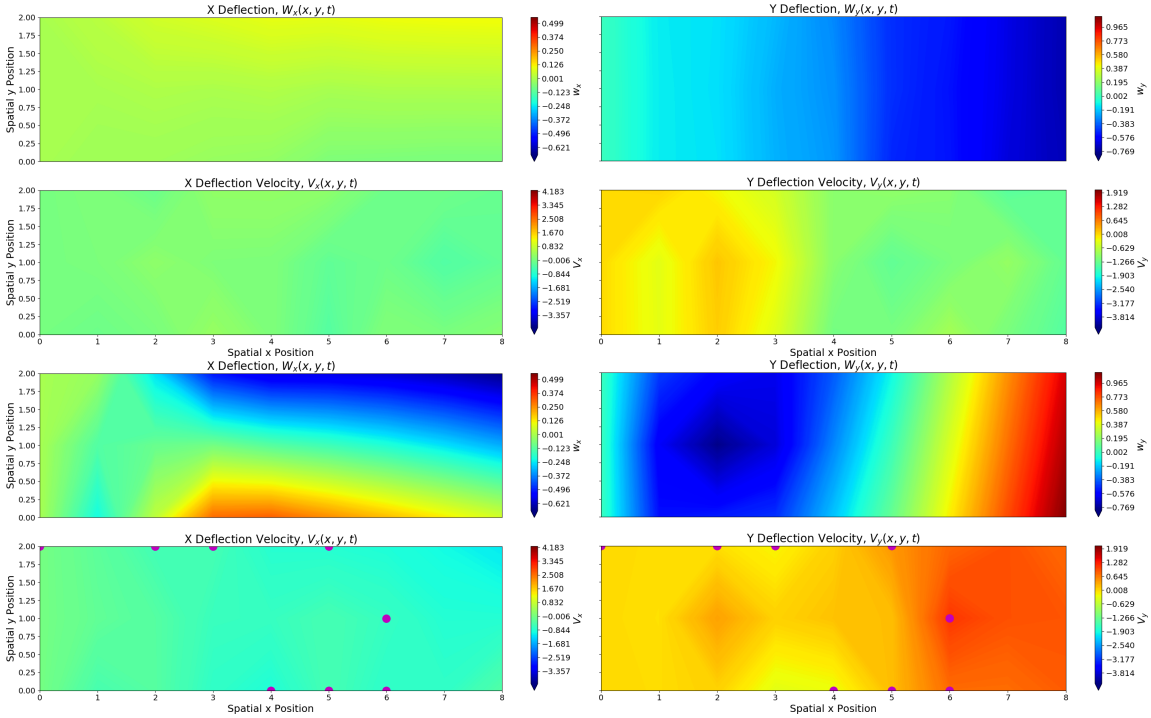


Figure 6.9: 2D Soft Arm Reaching Task contour plots of a random trajectory rollout at final time, where color represents magnitude. Purple circles represent actuators on the controlled system placed by actuator co-design optimization. (top left) uncontrolled final time snapshots of x deflection and x deflection velocity (top right) uncontrolled final time snapshots of y deflection and y deflection velocity (bottom left) controlled final time snapshots of x deflection and x deflection velocity (bottom right) controlled final time snapshots of y deflection and y deflection velocity. Videos of a random trajectory rollout of the controlled and uncontrolled system evolving in time can be found at https://youtu.be/yo48a6JqKE0.

The system is spatially discretized by lumping it into a Cartesian grid of point mass particles. Each particle, identified by horizontal index i and vertical index j, has a position $s_{i,j}$ and a velocity $v_{i,j}$. The resting length between adjacent particles is taken to be a constant length l. The strain divergence term $\text{div}(\sigma)$ in eq. (6.5) can be approximated by finite difference discretizations of the divergence operator and strain terms. A detailed derivation and discussion of this discretization scheme can be found in [131]. In this case the system has position and velocity states for each dimension over the 2D region, totaling 108 states. Temporal discretization is accomplished by solving the spatially discretized system with the explicit Euler method.

Just as in the case of the 1D Euler-Bernoulli SPDE, the actuation of this system enters

through the acceleration channels, as forces, yet the task is evaluated on the position channels. This results in an evident time-delay between actuation and effect in position space. The soft manipulator was initialized straight and level, with actuators initialized by sampling from a uniform distribution over $[0, w] \times [0, h]$, where w is the width of the system and h is the height.

The actuation scheme was chosen to mimic a class of manipulators found in biological systems such as cephalopod tentacles or elephant trunks known as muscular hydrostats [130], in which actuation of incompressible materials conserves the volume of the manipulator. We approximate this property by placing actuators on particles and adjusting the resting length of the tensile connections with adjacent particles. Positive control inputs to the actuator reduce the resting length of horizontal members and increase the resting length of vertical members by the same amount. Negative control inputs have the corresponding opposite effects on the resting length of adjacent members. For small actuator inputs, the volume of the manipulator is approximately conserved, resulting in behaviors such as reduction in thickness as the manipulator is axially extended.

The policy network utilized a similar CNN architecture as in the 2D heat experiment, with the entire state space treated as the input. The CNN produces an action signal, which through the actuation model act in the velocity dynamics as forces. For this experiment, the convolutional layers of the policy network were initialized with a Xavier [134] initialization, while the fully-connected output layer was initialized with zeroes.

As described earlier, a positive control signal on an actuator contracts adjacent horizontal members and expands adjacent vertical members. Thus in order to deflect the tip upwards one unit, the actuators must act in concert to contract the top surface towards the root for vertical deflection, and utilize the bottom surface as needed to minimize the horizontal displacement. Depicted are contour plots of each state of the uncontrolled (top) and controlled (bottom) system at the final time, with optimized actuators depicted in magenta in the velocity states.

A video of the uncontrolled and controlled state evolution is available at the link provided [7]. As depicted, the algorithm successfully places actuators in pertinent locations, and is able to achieve the goal state even for exaggerated gravitational forces two orders of magnitude larger than "normal" for our simulation parameters.

Note that the algorithm co-located four of the ten actuators allocated to the algorithm during optimization, which may be interpreted as an attempt by the algorithm to reduce control effort. Such actuator co-location can lead to identification of the minimum set of actuators to achieve a task. The actuator placement in this experiment is quite interesting and merits further analysis. It is challenging to deduce the choices in actuator placement and control, however the authors conjecture that the top left actuator, which receives a negative control signal with the largest control signal magnitude, is mostly responsible for maintaining axial tip location, while the rest of the top surface actuators are responsible for contracting the surface for an upwards deflection.

These results demonstrate intelligent actuator placements that leverage the system dynamics. The proposed policy and co-design optimization approach is evidently applicable even for nonlinear, 2nd-order, stochastic, 2D PDE systems. The authors intend to continue to analyze and extend these results further.

6.7 Discussion

Each of the above experiments presents an interesting challenge from the perspective of policy optimization and co-design. The system domains, behavior, and dimensionality are quite varied, and are representative of many of the natural phenomena found in nature and robotics literature. In each of these cases, the optimization parameters are initialized in a random way, so that the optimization does not have a "warm start". The two fully connected and convolutional policy network architectures were rather simple and shallow, did not undergo significant hyper-parameter tuning, and were re-used in all 1D and all 2D

[7] Video: https://youtu.be/yo48a6JqKE0

experiments, respectively. In several of the experiments, optimized actuator locations did not coincide with a human a-priori placement, which in the case of the Nagumo equation resulted in outperformance of the human placement by the algorithm.

The proposed algorithm was successful at joint actuator and policy optimization in each of these cases. Actuator placement optimization through the proposed framework appeared to leverage the dynamics in all of the provided experiments. In practice, the authors found that the presented loss function is equipped with several useful properties. Firstly, it incorporates significant information density, which allows numerous back-propagation gradient paths for both actuator co-design and policy optimization. Secondly, the expectation over rollouts allows a form of exploration of the state space. Finally, exponentiated weighting of trajectory performance allows the loss to clearly differentiate between better state trajectories and worse state trajectories. Indeed a larger quantity of rollouts over a system with Cylindrical Wiener noise plays a useful role in the proposed sampling-based optimization scheme, however most experiments were successful with only 50 rollouts.

The intuition behind the joint policy and actuator co-design optimization presented in this chapter as opposed to alternating optimization, which is often used for control and co-design optimization, may be explained as follows. As stated in section 6.3, the proposed approach develops a loss function defined in the path integral sense, and is evaluated by trajectory rollouts of the controlled system. At each time step of each trajectory rollout, the system evolution simultaneously depends on the policy parameters and the co-design parameters. It is evident that the specific loss function being used in the proposed approach has simultaneous gradient information for *all* design variables; there is not a separate loss function for policy network performance and a separate loss function for actuator design performance.

As such, an alternating approach must either a) only use the dense loss information for policy updates on the inner loop or b) collect gradients with respect to actuator design variables on the inner loop for a large outer loop update. Each option presents obvious

issues. The first in essence "wastes" the loss information for a potential actuator design update, while the second potentially sums conflicting gradient information. Instead, the proposed joint optimization approach *leverages* the dense information in the loss function to simultaneously update *all* design variables, where update rates can be simply controlled by the respective learning rates. Thus, one may still prevent actuator variables from updating "too fast" compared to the rate of improvement of the policy variables, or vice versa.

One may also understand the advantages of joint optimization by viewing the trajectory rollouts as a connected graph. Here, each time step may be viewed as a layer of a dynamics network that depends on the trainable parameters $\mathbf{x}$ and Θ. In this context, the loss function may be also viewed as forming a connected graph between the loss in eq. (6.41) and the trainable parameters $\mathbf{x}$ and Θ. As such, the policy and co-design variables are akin to parameters of a RNN as they are fixed in time. Thus the question of joint optimization vs alternating optimization reduces to the much simpler and obvious question of joint vs. alternating optimization of parameters of a neural network, which are by default optimized in a joint optimization mode.

However, the STSO optimization framework is not without fault. Throughout our experiments, RAM usage grows with actuation variables, problem size, time horizon, and the number of policy variables. In the context of our path integral graph, these scalability issues are similar to the scalability issues that arise in very deep, very wide neural architectures. As we continue to scale further, RAM usage may be a limiting factor.

One approach to tackle memory complexity is through accelerated variants of ADMM [135], which have become a popular method to tackle large scale optimization. Such methods split memory allocation by considering an alternating optimization problem, which will likely require substantially more iterations than joint optimization for convergence. Orthogonally, Sparse Neural Networks (SNNs) are a recent neural architecture that is of growing popularity for image classification tasks due to substantially lower Floating Point Operations (FLOPs) and substantially lower memory requirements for each inference [136, 137, 138].

Paired with sparse discretizations of PDEs, as in [139], and the *SparseForwardPass* method developed in this chapter for sparse evaluation of policy integrals, it may be possible to construct a completely sparse path integral graph. Such approaches are appealing future research directions to tackle scalability to 3D problem spaces, longer time horizons, deeper networks, and larger sets of actuator co-design optimization variables.

6.8 Conclusion

This chapter presents a measure-theoretic policy and actuator co-design optimization framework, which was developed in Hilbert spaces for the control of stochastic partial differential equations. Necessary mathematical results for extension to second-order SPDEs were presented and proved. Novel methods were introduced to decrease computational complexity and increase optimization performance. The resulting path integral loss function was optimized with a popular variant of gradient descent, and the optimization architecture was applied to six simulated SPDE experiments that each exhibited unique challenges. The last of which is a biologically inspired model of soft-robotic manipulators with muscular-like actuation, which connects to our goals of establishing capabilities for the further development of soft-body robotics. The results demonstrate that the proposed approach can perform joint policy and actuator co-design optimization on varied complex nonlinear stochastic PDE systems.

The presented approach is a new way of performing optimization and can lead to many applications in soft robotics, soft materials, morphing, and continuum mechanics. The results are encouraging to the authors. We plan to continue to develop methods for scalability, explore actuator shape optimization, investigate novel soft-robotic models, and apply the approach to a variety of hard spatio-temporal problems traditionally outside of the realm of robotics research.

Part II

Control Optimization for Spatio-Temporal Systems in Quantum Mechanics

CHAPTER 7
INTRODUCTION AND BACKGROUND

Control in the classical domain is generally split into two main categories: open-loop and closed-loop control. Open-loop control predicts system states and prescribes a policy that will best steer the state based on the prediction of either instantaneous state transitions or expected state trajectories. In this case the control may be explicitly a function of the predicted state, but by design does not incorporate true state information. The closed-loop control paradigm on the other hand is based on an assumption that we can infer the true state information from some measured system output, and use that as feedback to alter a control policy. Thus in this case the control explicitly depends on the true state of the system, and can therefore "react" to system behavior that may not have been predicted through state observation.

The currently dominant paradigm for optimal control in the quantum domain is one of open-loop control. The dominant methods are based on the Pontryagin Maximum Principle, as in [33, 140, 141, 142, 143, 144, 145, 146], the Krotov principle of optimality[1], as in [148, 149, 150, 151, 152, 153, 154, 34, 155, 156] , the Gradient Ascent Pulse Engineering (GRAPE) method, as in [157, 158], and recently through RL in [35, 159]. These methods have demonstrated many interesting results, yet are quite restrictive. An effect of this paradigm may be the rather limited state of the art coherence times, which are typically on the order of minutes [160], and are largely the result of dramatic engineering efforts focused on complete system isolation.

System isolation has become a significant effort in recent methods due to the consideration of the system as a closed system in the vast majority of recent methods. The closed system assumption by its very nature assumes, and thus requires, that the system is in

[1] see [147] for an introduction and review of Krotov methods

complete isolation. This is an assumption that cannot be guaranteed in general. Methods that consider interactions of the system with its environment are known as open quantum systems and are typically much more challenging from a control perspective. Some of the above methods have been extended to the open quantum paradigm, as in [161, 162]. The interested reader can refer to [163] for a review of control methods for open quantum systems as of 2016.

The performance of control methods may be most pertinent in the context of coherence control and state preparation in quantum computing, where one seeks to reach and stabilize some quantum state of qubits. Similar quantum control problems exist in other forms, including optimal quantum gate synthesis[164, 165, 166, 159, 167, 168, 169], where one seeks to provide a unitary transformation, or gate, which transforms an initial qubit state into a desired qubit state. This can equivalently also be viewed as a control problem. These gates form the fundamental building blocks of the quantum computing paradigm, wherein a Non-Polynomial-Time (NP) problem embedding is directed by quantum gates towards minima that solve the encoded problem. In this context, the efficacy of the quantum computing paradigm, and thus the goal of quantum supremacy, is tied to the assumptions in the evolution equations of the underlying system and the control solutions that take a system from one state to another.

Shortcomings in coherence stabilization and gate synthesis can be seen as shortcomings of control paradigms, and have led to major efforts to perform computation in the presence of "noisy" quantum states, i.e. where the state is moving towards decoherence or has a large uncertainty. Such efforts in error-correcting codes lead to a large computational overhead dedicated to error correction. However, better control solutions may lead to computing paradigms that dedicate fewer qubits to error correction [170], leading to quantum computing architectures with lower overhead for arbitrarily large qubit systems.

7.1 Dynamics of Open Quantum Systems and QND Measurement

Let $\mathscr{H}_S$ denote the separable Hilbert space of the closed system. In a closed quantum system, one can write the state evolution of a quantum density operator $\rho_S \in \mathscr{X}(\mathscr{H}_S)$, where $\mathscr{X}(\mathscr{H}_S)$ is the space of density operators on $\mathscr{H}_S$. The state evolution is given by the Liouville-von Neumann equation

$$\frac{\mathrm{d}}{\mathrm{d}t}\rho_S(t) = \frac{-i}{\hbar}[H_S(t), \rho_S(t)] \tag{7.1}$$

where $H_S(t)$ is a time-varying Hermitian operator defined as the system Hamiltonian, and $[\cdot,\cdot]$ is the commutator between two operators.

An open quantum system is a closed quantum system S coupled to another system B, called the environment. The total system can be expressed as $S+B$, which is itself another closed system. Instead of just including the quantum system S, the closure now includes the system and an environment B which S interacts with. The system S has potentially infinite degrees of freedom and is referred to as the “reduced system", while an environment B with infinite degrees of freedom is referred to as a “reservoir". A reservoir in equilibrium is referred to as a “heat bath" or simply a “bath". The total Hilbert space of $S+B$ is given by $\mathscr{H} = \mathscr{H}_S \otimes \mathscr{H}_B$, with time-dependent Hamiltonian [171]

$$H(t) = H_S \otimes I_B + I_S \otimes H_B + H_I(t). \tag{7.2}$$

Observables of the system S are always realized as operators of the form $\hat{A} \otimes \hat{I}_B$. Therefore, the expectation of the observable A of system S is given by $\langle A \rangle = \mathrm{Tr}_S\{A\rho_S\}$, where Tr_S is the partial trace with respect to a complete orthonormal basis in S (i.e the degrees of freedom of S), and $\rho_S = \mathrm{Tr}_B\rho$ is the reduced density matrix of S. The reduced density

matrix has evolution given by [171]

$$\rho_S(t) = \mathrm{Tr}_B\{U(t,0)\rho(0)U^\dagger(t,0)\}, \tag{7.3}$$

where $U(t,0)$ is the time evolution operator of the coupled system and the total density matrix is a composition of the system and environment density matrices $\rho(t) = \rho_S \otimes \rho_B$. We can similarly apply the partial trace to the Liouville-von Neumann equation

$$\frac{\mathrm{d}}{\mathrm{d}t}\rho_S = \frac{-i}{\hbar}\mathrm{Tr}_B\big[H(t), \rho(t)\big]. \tag{7.4}$$

After assuming markovian dynamics so that we can find a dynamical semigroup, and several steps that can be found in the Appendix, one obtains the Lindblad Master Equation with normalized $\hbar$ [171]

$$\frac{\mathrm{d}}{\mathrm{d}t}\rho_S = \mathscr{L}\rho_S = -i[H,\rho_S] + \sum_{k=1}^{N^2-1}\gamma_k\Big(A_k\rho_S A_k^\dagger - \frac{1}{2}A_k^\dagger A_k\rho_S - \frac{1}{2}\rho_S A_k^\dagger A_k\Big). \tag{7.5}$$

where $\{A_m\}$ forms an orthonormal basis of all Hilbert-Schmidt operators on $\mathscr{H}_S$. The Linbdblad form in eq. (7.5) is often written compactly as

$$\frac{\mathrm{d}}{\mathrm{d}t}\rho_S = -i[\hat{H},\rho_S] - \frac{1}{2}\big[A,[A,\rho_S]\big]. \tag{7.6}$$

This can be further simplified to

$$\mathrm{d}\rho_S = \mathscr{L}_0\rho_S\mathrm{d}t + \mathscr{D}[A]\rho_S\mathrm{d}t, \tag{7.7}$$

where $\mathscr{L}_0\rho_S := -i[H,\rho_S]$ and $\mathscr{D}[A]\rho_S := -\frac{1}{2}\big[A,[A,\rho_S]\big]$.

Let us consider again the closed-loop (which should not be confused with closed system) control approach, where we continuously feedback system measurements to yield a controller that is *reactive* to the true state evolution. In the closed quantum system, measurements are

known to collapse quantum systems into classical states, thus feedback has very limited feasibility. However, there exists a measurement paradigm that does not lead to instantaneous collapse, known as Quantum Non-Demolition (QND) measurement [8, 172, 173]. Let $A : \mathscr{H}_S \to \mathscr{H}_S$ denote some system observable then a QND observable requires that

$$[A(t), A(s)] = 0, \quad \forall\, t, s \in [0, T] \tag{7.8}$$

where the time evolution of the system has been absorbed into the observable (Heisenberg or Interaction picture) as

$$A(t) = U^{\dagger}(t) A U(t) \tag{7.9}$$

for some unitary evolution operator $U(t)$ that evolves the state $|\psi(0)\rangle$ into $|\psi(t)\rangle$. A more detailed explanation can be found in [174].

The existence of a measurement process that does not collapse the quantum system state begs the question: can this measurement be used for a feedback process? This question was addressed in numerous works by V.P. Belavkin [8, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189], and many refer to the resulting equations as Belavkin Equations. The first step to this end is to mathematically describe the "back-action" effect of a QND measurement process on a quantum dynamical system. The work by V.P. Belavkin consider this phenomena, and present a jump process back-action for discrete quantum measurements, and a Brownian processes back-action for continuous quantum measurements [179]. The discrete equations are derived in the appendix, and the continuous equations are found by a limiting process [188]. The discrete measurement case has the form

$$\mathrm{d}\rho_c^t = \mathscr{L}_0 \rho_c^t \mathrm{d}t + \mathscr{D}[V]\rho_c^t \mathrm{d}t + \left(\frac{V \rho_c^t V^*}{\mathrm{Tr}[V \rho_c^t V^*]} - \rho_c^t \right) \mathrm{d}M_t \tag{7.10}$$

$$\text{Innovation:} \quad \mathrm{d}M_t = \mathrm{d}N_t - |\kappa_s|^2 \mathrm{Tr}[V \rho_c^t V^*] \mathrm{d}t \tag{7.11}$$

and the continuous measurement case has the form

$$\mathrm{d}\rho_c^t = \mathscr{L}_0\rho_c^t\mathrm{d}t + \mathscr{D}[V]\rho_c^t\mathrm{d}t + \Big(V\rho_c^t + \rho_c^t V^\dagger - \mathrm{Tr}\big[(V+V^\dagger)\rho_c^t\big]\rho_c^t\Big)\mathrm{d}W_t \tag{7.12}$$

$$\text{Innovation:} \quad \mathrm{d}W_t = \mathrm{d}y_t - \mathrm{Tr}\big[(V+V^\dagger)\rho_c^t\big]\mathrm{d}t \tag{7.13}$$

where $\mathrm{d}M_t$ and $\mathrm{d}N_t$ are Poisson processes and $\mathrm{d}W_t$ is a standard zero-mean Wiener process. $\mathrm{d}N_t$ has intensity $|\kappa_s|^2\mathrm{Tr}\big[V\rho_c^t V^*\big]$. Here ρ_c^t is the system density state conditioned on the measurement outcome. This conditioning is key; if we were to "throw out" the measurement process, we would average over the noise process ($\mathrm{d}N_t$ for discrete measurement, $\mathrm{d}W_t$ for continuous measurement) and thus recover the standard Lindblad master equation. Here, the system closure includes the system S and the measurement process R with operator V. One can similarly consider a closure that includes the system S, the environment B and the measurement process R, However this is omitted for brevity.

Now eq. (7.12) is a quantum SPDE with quantum unconditional evolution governed by $\mathscr{L}_0\rho_c^t$ and standard Wiener process $\mathrm{d}W_t$ in the classical sense [66]. It is interesting to note that one can draw parallels between eq. (7.12) and the Kushner-Stratonovich SPDE [190]. This is shown in Appendix Q. Just as in the case of the Kushner-Stratonovich SPDE, the stochasticity is the result of conditioning on the measurement process $\mathrm{d}y_t$. Following this logic, one can think of the Belavkin equation as a partially observable problem.

7.2 Optimal Control Theory for Open Quantum Systems with QND Measurement

With the continuous measurement process that yields the quantum SDE given by the Belavkin equation eq. (7.12), we return to the question of whether this can lead to the application of standard methods from optimal control theory in the classical regime. In order to address this, we must first establish some sort of actuation on the system. In many experimental setups, this is achieved by a controlled Hamiltonian H_u that is typically a potential function

generated by a coupled magnetic field. The controlled quantum SPDE takes the form

$$\mathrm{d}\rho_c^t = \mathscr{L}_0\rho_c^t\mathrm{d}t + \mathscr{D}[V]\rho_c^t - i\sum_j u_{t,j}[H_{u,j},\rho_c^t]\mathrm{d}t + B(\rho_c^t)\mathrm{d}W_t \tag{7.14}$$

where we have defined $B(\cdot)$ for simplicity as $B(\rho) := V\rho + \rho V^\dagger - \mathrm{Tr}\Big[(V+V^\dagger)\rho\Big]\rho$.

The quantum optimal control problem for open quantum systems with QND measurement was first explored in several works by V.P. Belavkin and L. Bouten [8, 187, 189], with modern efforts by H. Mabuchi and J.K. Stockton in [190, 191, 192]. Define the total averaged cost (or expected cost) as

$$J\big(\rho_c(t),\mathbf{u}(t),t\big) = \left\langle \int_0^T L\big(\rho_c(t),\mathbf{u}(t),t\big)dt + L_f\big(\rho_c(T),T\big)\right\rangle \tag{7.15}$$

where L is the running cost, L_f is the terminal cost, and $\mathbf{u}(t)$ is the vector of the control signals u_j. Next define the value functional as

$$V\big(\rho_c(t),t\big) := \min_{\mathbf{u}(t)} J\big(\rho_c(t),\mathbf{u}(t),t\big). \tag{7.16}$$

Application of the Bellman principle of optimality leads to the quantum HJB equation [190]

$$-\frac{\partial}{\partial t}V\big(\rho_c(t),t\big) = \min_{\mathbf{u}(t)}\left[L(\rho_c(t),\mathbf{u}(t),t) + \Big\langle F\big(\rho_c(t),\mathbf{u}(t),t\big),\nabla V\big(\rho_c(t),t\big)\Big\rangle\right] \tag{7.17}$$

where F is the drift of the dynamics, given by

$$F\big(\rho_c(t),\mathbf{u}(t)\big) := \mathscr{L}_0\rho_c(t) + \mathscr{D}[V]\rho_c(t) - i\mathbf{u}^\top[H_\mathbf{u},\rho_c(t)] \tag{7.18}$$

and in this case the transpose operation is defined with respect to control degrees of freedom.

CHAPTER 8

VARIATIONAL OPTIMIZATION-BASED QUANTUM FEEDBACK CONTROL FOR OPEN QUANTUM SYSTEMS

In chapter 4, we leverage known connections between the information theoretic approach derived therein and the HJB equation. From this information theoretic perspective, we derive a feedback architecture for SPDEs in the classical regime. The key ingredients of the prior information theoretic architecture are 1) the Legendre transformation (free energy-relative entropy relationship), 2) the Gibbs optimal measure minimizer of the Legendre transformation, 3) definition of a variational optimization problem with respect to the Gibbs measure, and 4) the change of measures, or RN derivative, provided by a Girsanov Theorem for SPDEs. The resulting control framework arises by application of a Newton style analytical minimization to the resulting optimization problem, which is quadratic in $\mathbf{u}$ due to the quadratic nature of the change of measures.

One can explore a similar framework for closed-loop feedback control of a QND measurement of an open quantum system and apply it to realistic simulated experiments. The first ingredient, the Legendre transformation for QND measurement of open quantum systems was explored in [193], and indeed it yields a Gibbs measure minimizer identical to the classical case. Let $\tilde{\mathscr{L}}$ denote the measure of the controlled system in eq. (7.14), let $\mathscr{L}$ denote the measure of the uncontrolled analog of eq. (7.14), and let $J = J(\rho_t)$ denote an arbitrary state cost functional. Then the quantum Legendre transformation is given by

$$-\frac{1}{r}\log \mathbb{E}_{\mathscr{L}}\left[\exp(-rJ)\right] = \min_{\mathscr{U}(\cdot,\cdot)}\left[\mathbb{E}_{\tilde{\mathscr{L}}}(J) + \frac{1}{r}S_c(\tilde{\mathscr{L}}\,||\,\mathscr{L})\right], \tag{8.1}$$

where $r \in \mathbb{R}$ and for absolutely continuous measures $\tilde{\mathscr{L}} >> \mathscr{L}$, the relative entropy is given

by [66]

$$S_c(\tilde{\mathscr{L}}||\mathscr{L}) := \int_\Omega \ln \frac{\mathrm{d}\tilde{\mathscr{L}}(\mathbf{u})}{\mathrm{d}\mathscr{L}} \mathrm{d}\tilde{\mathscr{L}}(\mathbf{u}). \tag{8.2}$$

The minimizing measure of eq. (8.1) is given by the Gibbs measure

$$\mathrm{d}\mathscr{L}^* = \frac{\exp(-rJ)\mathrm{d}\mathscr{L}}{\mathbb{E}_{\mathscr{L}}\big[\exp(-rJ)\big]}. \tag{8.3}$$

Thus we can define the variational optimization problem using the KL divergence as

$$\mathbf{u}^* = \underset{\mathbf{u}}{\operatorname{argmin}} D_{KL}\big(\mathscr{L}^*||\tilde{\mathscr{L}}(\mathbf{u})\big) \tag{8.4}$$

$$= \underset{\mathbf{u}}{\operatorname{argmin}} \int_\Omega \ln\left(\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}(\mathbf{u})}\right) \frac{\mathrm{d}\mathscr{L}^*}{\mathrm{d}\mathscr{L}} \frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}(\mathbf{u})} \mathrm{d}\tilde{\mathscr{L}}(\mathbf{u}) \tag{8.5}$$

where the intermediate steps are the result of similar operations performed in chapter 4. Now it is quite clear that one needs the change of measures, or RN derivative, for QND-measured open quantum systems. To the best knowledge of the author, a change of measures for a change of drift does not previously exist in literature. However, in [194], the authors derive a change of measures for a change of diffusion. In the Appendix, we present a derivation of the change of measures, or RN derivative, based on a change of drift, which is given by

$$\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}(\mathbf{u})} = \exp\left(\int_0^T \mathbf{u}^\top \mathrm{Tr}_\rho\Big\{i\big[\mathbf{H}_u, \rho_s\big]B(\rho_s)^{-1}\mathrm{d}W_s\Big\} + \frac{1}{2}\int_0^T \mathbf{u}^\top \mathrm{Tr}_\rho\Big\{\big[\mathbf{H}_u, \rho_s\big]B(\rho_s)^{-2}\big[\mathbf{H}_u^\top, \rho_s\big]\Big\}\mathbf{u}\,\mathrm{d}s\right). \tag{8.6}$$

Plugging eq. (8.6) and the Gibbs measure into eq. (8.5) yields

$$\mathbf{u}^* = \underset{\mathbf{u}}{\operatorname{argmin}} \mathbb{E}_{\tilde{\mathscr{L}}}\left[\left(-\int_0^T \mathbf{u}^\top \mathrm{Tr}_\rho\Big\{i\big[\mathbf{H}_u, \rho_s\big]B(\rho_s)^{-1}\mathrm{d}W_s\Big\} - \frac{1}{2}\int_0^T \mathbf{u}^\top \mathrm{Tr}_\rho\Big\{\big[\mathbf{H}_u, \rho_s\big]B(\rho_s)^{-2}\big[\mathbf{H}_u^\top, \rho_s\big]\Big\}\mathbf{u}\,\mathrm{d}s\right)\frac{\exp(-r\tilde{J})}{\mathbb{E}_{\tilde{\mathscr{L}}}\big[\exp(-r\tilde{J})\big]}\right],$$

where

$$\tilde{J} = J + \int_0^T \mathbf{u}^\top \mathrm{Tr}_\rho \Big\{ i\big[\mathbf{H}_u, \rho_s\big] B(\rho_s)^{-1} \mathrm{d}W_s \Big\} + \frac{1}{2} \int_0^T \mathbf{u}^\top \mathrm{Tr}_\rho \Big\{ \big[\mathbf{H}_u, \rho_s\big] B(\rho_s)^{-2} \big[\mathbf{H}_u^\top, \rho_s\big] \Big\} \mathbf{u} \, \mathrm{d}s. \tag{8.7}$$

Since we apply control in discrete time, we approximate both terms as follows

$$\int_0^T \mathrm{Tr}_\rho \Big\{ \big[\mathbf{H}_u, \rho_s\big] B(\rho_s)^{-2} \big[\mathbf{H}_u^\top, \rho_s\big] \Big\} \mathbf{u} \, \mathrm{d}s \Big) \approx \sum_{l=1}^{L-1} \int_{t_l}^{t_{l+1}} \mathrm{Tr}_\rho \Big\{ \big[\mathbf{H}_u, \rho_s\big] B(\rho_s)^{-2} \big[\mathbf{H}_u^\top, \rho_s\big] \Big\} \mathrm{d}s \, \mathbf{u}_l$$

$$\int_0^T \mathrm{Tr}_\rho \Big\{ \big[\mathbf{H}_u, \rho_s\big] B(\rho_s)^{-1} \mathrm{d}W_s \Big\} \approx \sum_{l=1}^{L-1} \int_{t_l}^{t_{l+1}} \mathrm{Tr}_\rho \Big\{ \big[\mathbf{H}_u, \rho_s\big] B(\rho_s)^{-1} \mathrm{d}W_s \Big\}$$

Solving for the optimal control via Newton minimization yields the iterative update

$$\mathbf{u}_l^* = -\,\mathbb{E}_{\mathscr{D}} \left[\int_{t_l}^{t_{l+1}} \mathrm{Tr}_\rho \Big\{ \big[\mathbf{H}_u, \rho_s\big] B(\rho_s)^{-2} \big[\mathbf{H}_u^\top, \rho_s\big] \Big\} \mathrm{d}s \frac{\exp(-r\tilde{J})}{\mathbb{E}_{\mathscr{D}}\big[\exp(-r\tilde{J})\big]} \right]^{-1} \mathbb{E}_{\mathscr{D}} \left[\int_{t_l}^{t_{l+1}} \mathrm{Tr}_\rho \Big\{ i\big[\mathbf{H}_u, \rho_s\big] B(\rho_s)^{-1} \mathrm{d}W_s \Big\} \frac{\exp(-r\tilde{J})}{\mathbb{E}_{\mathscr{D}}\big[\exp(-r\tilde{J})\big]} \right] \tag{8.8}$$

where the inverse term is a matrix in the dimensionality of $\mathbf{u}_l$, but a scalar in the dimensionality of ρ_t. Likewise the second expectation term is a vector in the dimensionality of $\mathbf{u}_l$, but a scalar in the dimensionality of ρ_t. The main concern with this approach is the case in which $B(\rho_s)$ is singular. This can emerge in many experimental setups. Below we elucidate the singularity phenomena with two common experiments.

8.1 Two Qubit System

Consider the two-qubit quantum system given by the SME [195]

$$\begin{aligned} \mathrm{d}\rho_t = &-iu_1(t)[\sigma_y^{(1)}, \rho_t]\mathrm{d}t - iu_2(t)[\sigma_y^{(2)}, \rho_t]\mathrm{d}t - \frac{1}{2}\big[F_z, [F_z, \rho_t]\big]\mathrm{d}t \\ &+ \sqrt{\eta}\Big(\{F_z, \rho_t\} - 2\mathrm{Tr}(F_z\rho_t)\rho_t\Big)\mathrm{d}W_t, \end{aligned} \tag{8.9}$$

where $u_j(t)$ are two time-varying magnetic fields coupled to the two atoms, $F_z := \sigma_z^{(1)} \otimes I^{(2)} + I^{(1)} \otimes \sigma_z^{(2)}$ defines the coupling between the cavity and the electromagnetic field

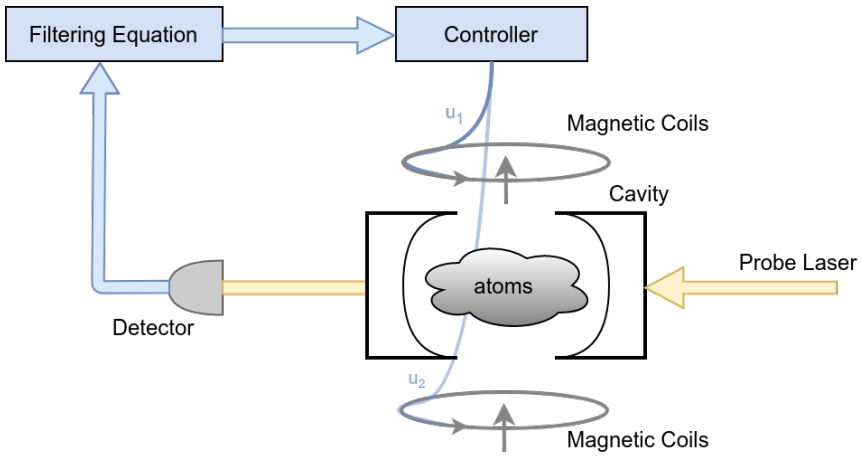


Figure 8.1: A graphical representation of a non-demolition measurement experiment of a cavity of atoms weakly coupled to a probe laser that are controlled by a magnetic field potential. This figure was adapted from [195].

produced by the probe laser, as depicted in fig. 8.1, and as usual $[\cdot,\cdot]$ and $\{\cdot,\cdot\}$ are the commutator and anti-commutator, respectively. We can simplify this equation by defining the following superoperators

$$B(\rho_t) := \sqrt{\eta}\Big(\{F_z,\rho_t\} - 2\mathrm{Tr}(F_z\rho_t)\rho_t\Big) \tag{8.10}$$

$$F(\rho_t) := -\frac{1}{2}\big[F_z,[F_z,\rho_t]\big] \tag{8.11}$$

Also, note that in this system $H = 0$, and $H_{u,j} = \sigma_y^{(j)}$. This yields

$$\mathrm{d}\rho_t = F(\rho_t)\mathrm{d}t - i\sum_j^2 u_j(t)\big[H_{u_j},\rho_t\big]\mathrm{d}t + B(\rho_t)\mathrm{d}W_t, \tag{8.12}$$

The main consideration is under what conditions the superoperator $B(\rho_t)$ is invertible. In cases where $B(\rho_t)$ is singular, the change of measures in eq. (8.6) is not well defined. In this experimental setup, one typically performs coherence control, wherein the goal is to

stabilize one of the four maximally entangles qubit states, known as Bell states

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}\Big(|\downarrow_1\downarrow_2\rangle + |\uparrow_1\uparrow_2\rangle\Big) \tag{8.13}$$

$$|\Phi^-\rangle = \frac{1}{\sqrt{2}}\Big(|\downarrow_1\downarrow_2\rangle - |\uparrow_1\uparrow_2\rangle\Big) \tag{8.14}$$

$$|\Psi^+\rangle = \frac{1}{\sqrt{2}}\Big(|\downarrow_1\uparrow_2\rangle + |\uparrow_1\downarrow_2\rangle\Big) \tag{8.15}$$

$$|\Psi^+\rangle = \frac{1}{\sqrt{2}}\Big(|\downarrow_1\uparrow_2\rangle - |\uparrow_1\downarrow_2\rangle\Big) \tag{8.16}$$

With the above experimental setup, one can easily find that the term $B(\rho_t)$ is a singular matrix for each of the Bell states.

8.2 Dissapative Homodyne Detection

The Homodyne detection experiment was among the first non-demolition measurement experiments [173], and can be viewed from the photon counting (jump noise) or continuous diffusion (brownian noise) cases. In this experiment a cavity system emits photons when the atoms in the cavity are excited. The photon leakage is mixed with a local oscillator of the same frequency, and the mixed beam is then detected. The experimental setup is depicted in [196].

The dissipative Homodyne detection experiment is given in Fock space by the SME

$$\begin{aligned} \mathrm{d}\rho_t = &-i[H_0,\rho_t]\mathrm{d}t - i\sum_j u_j[H_{u_j},\rho_t]\mathrm{d}t - \frac{1}{2}\sqrt{1-\eta}\sqrt{\gamma}\big[a,[a,\rho_t]\big]\,\mathrm{d}t \\ &+\sqrt{\eta}\sqrt{\gamma}\Big(\{a,\rho_t\} - 2\mathrm{Tr}(a\rho_t)\rho_t\Big)\mathrm{d}W_t \end{aligned} \tag{8.17}$$

where H_0 is the typical unforced Hamiltonian of the quantum harmonic oscillator, a is the usual annihilation operator, and $\mathbf{H}_u$ is the Hamiltonian of the external forcing, in this case

provided by a coupled electromagnetic field. Defining

$$F(\rho_t) := -i[H_0, \rho_t] - \sqrt{1-\eta}\sqrt{\gamma}\mathscr{D}[a]\rho_t \tag{8.18}$$

$$B(\rho_t) := \sqrt{\eta}\sqrt{\gamma}\{a, \rho_t\} - 2\mathrm{Tr}(a\rho_t)\rho_t \tag{8.19}$$

we again have the form in eq. (8.12). In this case we again find that the $B(\rho_t)$ is singular in many regions of the state space.

8.3 Conclusion

In this chapter, we followed the approach of chapter 4 in order to develop an iterative update scheme for feedback control optimization of a SME conditioned on a weak QND measurement. Starting from the free energy-relative entropy relationship in ITC, we developed an optimization problem with respect to the Gibbs minimizing measure. Mathematical tools such as the change of measures, or RN derivative, were applied in order to realize an iterative update scheme that depends on the non-singularity of the $B(\rho)$ operator.

After attempting to apply the iterative scheme to a number of common open quantum experiments with QND measurement, it was observed that in several critical parts of the state space, for several control setups, the $B(\rho)$ operator becomes singular. Despite the success of similar approaches for classical SPDEs, this is identified as a critical shortcoming in the context of open quantum systems with QND measurement. In the subsequent chapters, we apply the lessons learned here, and we develop several methods that avoid the fundamental problems that arise in this chapter. Thus the author considers this "null" result to have importance in the overall goal of developing novel quantum control methodologies for SMEs.

CHAPTER 9

STOCHASTIC OPTIMIZATION FOR LEARNING QUANTUM FEEDBACK CONTROL

In light of the singularities of the $B(\rho_t)$ that arise in a number of natural experimental setups for open quantum systems with QND measurement, we seek methods that can bypass the change of measures. The Gradient-based Adaptive Stochastic Search (GASS) method was introduced in [113], and has been shown in [197] to generalize the information-theoretic approach in various ways. This approach has provable convergence characteristics, and offers generality and flexibility.

Consider the two qubit stochastic master eq. (8.9) equation in the general simplified form

$$\mathrm{d}\rho_t = F(\rho_t)\mathrm{d}t + G(\rho_t, \mathbf{u}_t)\mathrm{d}t + B(\rho_t)\mathrm{d}W_t \tag{9.1}$$

where $G(\rho_t, \mathbf{u}_t)$ is a state dependent control term. In the above example problem $G(\rho_t, \mathbf{u}_t)$ takes the form $G(\rho_t, \mathbf{u}_t) = \sum_i^2 u_i[\sigma_y^{(i)}, \rho_t]$, however in a more general N-qubit experiment, one may require all single-particle Pauli matrices. Thus $G(\rho_t, \mathbf{u}_t)$ takes the form

$$G(\rho_t, \mathbf{u}_t) = \sum_{i,j=1}^{N,3} u_{ij}\left[\sigma_j^{(i)}, \rho_t\right] \tag{9.2}$$

where $\sigma_j^{(i)}$, $j \in 1,2,3$ denote single particle Pauli matrices of each axis x, y, z. Despite appearing the context of qubit systems, the form of eq. (9.1) is quite general, and can represent virtually *any* open quantum system with continuous QND measurement.

Consider the task of reaching some target state ρ_{des}, as measured by some general cost metric $J(\rho_t, \mathbf{u}_t)$. The minimizing control is most generally expressed by the following path

integral optimization problem

$$\mathbf{u}^* = \underset{\mathbf{u}\in\mathscr{U}}{\operatorname{argmin}}\, \mathbb{E}_Q\left[J(\rho,\mathbf{u})\right] \tag{9.3a}$$

$$\text{s.t.} \quad \mathrm{d}\rho_t = F(\rho_t)\mathrm{d}t + G(\rho_t,\mathbf{u}_t)\mathrm{d}t + B(\rho_t)\mathrm{d}W_t, \tag{9.3b}$$

where the expectation defines a path integral over controlled state trajectories with measure Q. The set $\mathscr{U}$ is the admissible set of controls and may impose constraints on the control. One may also include constraints on the state ρ, however these are omitted from this derivation for simplicity.

The performance functional $J : H^2 \times \mathbb{R}^m \to \mathbb{R}$ is some real-valued, potentially non-convex, discontinuous, and non-differentiable functional, which must be minimized. Such a function imposes many difficulties from the context of optimization theory and optimal control theory. In the GASS approach, we bypass these difficulties through stochastic approximation. Let $f(u;\theta)$ be a distribution belonging to the exponential family of distributions. Then the optimization problem is approximated as

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\, \mathbb{E}_{Q,f(\mathbf{u};\theta)}\left[J(\rho,\mathbf{u})\right] \tag{9.4a}$$

$$\text{s.t.} \quad \mathrm{d}\rho_t = F(\rho_t)\mathrm{d}t + G(\rho_t,\mathbf{u}_t)\mathrm{d}t + B(\rho_t)\mathrm{d}W_t, \tag{9.4b}$$

$$\mathbf{u}_t \sim f(\mathbf{u}_t;\theta) \tag{9.4c}$$

Furthermore, we introduce the smooth (continuously differentiable a.e.), non-increasing shape function $S : \mathbb{R} \to \mathbb{R}$ and the logarithm function to obtain the following modified

optimization problem

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \log \mathbb{E}_{Q,f(u;\theta)}\left[S\big(J(\rho,\mathbf{u})\big)\right] \tag{9.5a}$$

$$\text{s.t.} \quad \mathrm{d}\rho_t = F(\rho_t)\mathrm{d}t + G(\rho_t,\mathbf{u}_t)\mathrm{d}t + B(\rho_t)\mathrm{d}W_t, \tag{9.5b}$$

$$\mathbf{u}_t \sim f(\mathbf{u}_t;\theta) \tag{9.5c}$$

Solving this optimization problem with gradient-based parameter adaptation has been shown to have numerous appealing convergence characteristics detailed in [113], however a key observation is that this formulation does not incorporate the measurement from the measurement process dW and is a purely feed-forward control. In this representation, one may compare this framework to the popular feedforward frameworks such as GRAPE or Krotov for optimal control of quantum systems without feedback (c.f. the approaches in [158]), however, the goal in defining the SME in eq. (9.1) is to realize a *feedback* control optimization algorithm. In the following we consider a number of modifications to the above optimization problem to achieve this goal.

SME with stochastic actuators and dynamic feedback compensation.

Consider the optimization problem

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \log \mathbb{E}_{Q,U,f(\varphi;\theta)}\left[S\Big(J(\rho,\mathbf{u})\Big)\right] \tag{9.6a}$$

$$\text{s.t.} \quad \mathrm{d}\rho_t = F(\rho_t)\mathrm{d}t + G(\rho_t,\mathbf{u}_t)\mathrm{d}t + B(\rho_t)\mathrm{d}W_t, \tag{9.6b}$$

$$\mathrm{d}\mathbf{u}_t = G_u(\rho_t,\mathbf{u}_t;\varphi_1) + \Sigma \mathrm{d}V_t \tag{9.6c}$$

$$\mathbf{u}_{t_0} = G_0(\varphi_2) \tag{9.6d}$$

$$\varphi := [\varphi_1,\varphi_2] \sim f(\varphi;\theta), \tag{9.6e}$$

where Q is the measure of the controlled dynamics, U is the measure of the dynamic compensator, and $f(\varphi;\theta)$ is a distribution, parameterized by θ, which belongs to the exponential family of distributions. We include noise in the compensator to represent a realistic noisy digital compensation signal, however this can be neglected to reduce the sampling complexity. The function $J : H^2 \times \mathbb{R}^m \to \mathbb{R}$ is some real-valued, potentially non-convex and non-differentiable metric, and the function $S : \mathbb{R} \to \mathbb{R}$ is a smooth shape function. The function $G_u : H^2 \times \mathbb{R}^m \times \mathbb{R}^p$ is the drift of the dynamic compensator.

Here, we must approximate the expectation with finite samples from three processes, namely the original SME, the stochastic dynamic compensator, and the compensator initial condition distribution. This approach may enable substantially more exploration of the state space, however this comes at the cost of sampling *three* distributions, which can quickly become computationally expensive. One may notice that these compensator dynamics are functionally similar to a stochastic RNN. The deterministic RNN case will be explored later.

SME with linear parametric static feedback compensation.

Consider the optimization problem

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \log \mathbb{E}_{Q,f(\varphi;\theta)} \left[S\Big(J(\rho, \mathbf{u}) \Big) \right] \tag{9.7a}$$

$$\text{s.t.} \quad \mathrm{d}\rho_t = F(\rho_t)\mathrm{d}t + G(\rho_t, \mathbf{u}_t)\mathrm{d}t + B(\rho_t)\mathrm{d}W_t, \tag{9.7b}$$

$$\mathbf{u}_t = K_1(\varphi_1)\rho_t + K_2(\varphi_2) \tag{9.7c}$$

$$\varphi := [\varphi_1, \varphi_2] \sim f(\varphi;\theta) \tag{9.7d}$$

Under the realization that the controller in [195] is quite similar to a P-controller on the trace distance to the goal state, this has a static compensator with an explicit parametric linear feedback policy. The expectation in eq. (9.7a) is a double expectation composed of an expectation over the SME and an expectation over the exponential family.

Note that this control policy can be realized through a fully connected network with

ReLU activations as

$$\mathbf{u}_t = K(\rho_t; \varphi), \tag{9.8}$$

where $K : H^2 \to \mathbb{R}^m$ is the linear policy network. This motivates the use of nonlinear policy networks.

SME with nonlinear parametric static feedback compensation.

Consider the optimization problem

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \log \mathbb{E}_{Q, f(\varphi;\theta)} \left[S\Big(J(\rho, \mathbf{u}) \Big) \right] \tag{9.9a}$$

$$\text{s.t.} \quad \mathrm{d}\rho_t = F(\rho_t)\mathrm{d}t + G(\rho_t, \mathbf{u}_t)\mathrm{d}t + B(\rho_t)\mathrm{d}W_t, \tag{9.9b}$$

$$\mathbf{u}_t = \Phi(\rho_t; \varphi) \tag{9.9c}$$

$$\varphi \sim f(\varphi; \theta) \tag{9.9d}$$

where Φ is a nonlinear feedback policy parametrized by φ. This could be a FNN or a CNN, but in general simply represents a nonlinear function of ρ without explicit time-dependence.

SME with nonlinear parametric dynamic feedback compensation.

Consider the optimization problem

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \log \mathbb{E}_{Q, f(\varphi,\theta)} \left[S\Big(J(\rho, \mathbf{u}) \Big) \right] \tag{9.10a}$$

$$\text{s.t.} \quad \mathrm{d}\rho_t = F(\rho_t)\mathrm{d}t + G(\rho_t, \mathbf{u}_t)\mathrm{d}t + B(\rho_t)\mathrm{d}W_t, \tag{9.10b}$$

$$\mathbf{u}_{t_{k+1}} = \Phi_{RNN}(\rho_{t_k}, \mathbf{u}_{t_k}; \varphi) \tag{9.10c}$$

$$\varphi \sim f(\varphi; \theta) \tag{9.10d}$$

where Φ_{RNN} is an RNN (e.g. LSTM network). Incorporating time-dependence in the policy endows the compensator with "dynamics", and enables treatment of a larger class of

problems compared to a static compensator.

One may also apply a Neural Ordinary Differential Equation (NODE) network [198] in place of eq. (9.10c). Instead of specifying a discrete sequence of hidden layers, NODE networks parametrize the derivative of the hidden state using a neural network, and as a result demonstrate *constant* memory cost as a function of network depth, significantly lower training losses, and can handle time irregularity in the discretization scheme. In many cases, NODE networks outperform RNN networks, and are a closer representation to a deterministic version of the dynamic compensation approach in eq. (9.6c).

9.1 Quantum GASS Parameter Update

The GASS method was first derived in [113]. Here we derive the parameter update under a minimization problem instead of a maximization problem, and use the above notation. Start with the general optimization problem

$$\mathbf{u}^* = \underset{\mathbf{u}\in\mathscr{U}}{\operatorname{argmin}} J(\rho,\mathbf{u}) \tag{9.11}$$

where $\mathscr{U} \subseteq \mathbb{R}^n$ is a non-empty compact set, and $F : H \times \mathscr{U} \to \mathbb{R}$ is a real-valued, potentially non-convex, discontinuous, and non-differentiable function. We avoid the inherent difficulties in $F(\mathbf{u})$ by transforming the problem into an approximation where $\mathbf{u}$ is sampled from the distribution $f(\mathbf{u};\theta)$

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \int_{\mathscr{U}} J(\rho,\mathbf{u}) f(\mathbf{u};\theta)\mathrm{d}\mathbf{u} = \mathbb{E}_{f(\mathbf{u};\theta)}\big[J(\rho,\mathbf{u})\big] \tag{9.12}$$

We additionally add a shape function that is differentiable, non-increasing, and positive semi-definite $S(\cdot) : \mathbb{R} \to \mathbb{R}_+$, which transforms our minimization problem into a maximization

problem

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \ln \int_{\mathscr{U}} S\big(J(\rho,\mathbf{u})\big) f(\mathbf{u};\theta)\mathrm{d}\mathbf{u} = \ln \mathbb{E}_{f(\mathbf{u};\theta)}\Big[S\big(J(\rho,\mathbf{u})\big)\Big] \tag{9.13}$$

We perform gradient updates in order to update the parameters θ of the distribution $f(\mathbf{u};\theta)$, which is assumed to belong to the exponential family of distributions.

$$\nabla_\theta \ln \int_{\mathscr{U}} S\big(J(\rho,\mathbf{u})\big) f(\mathbf{u};\theta)\mathrm{d}\mathbf{u} = \frac{\nabla_\theta \int_{\mathscr{U}} S\big(J(\rho,\mathbf{u})\big) f(\mathbf{u};\theta)\mathrm{d}\mathbf{u}}{\int_{\mathscr{U}} S\big(J(\rho,\mathbf{u})\big) f(\mathbf{u};\theta)\mathrm{d}\mathbf{u}} \tag{9.14}$$

$$= \frac{\int_{\mathscr{U}} S\big(J(\rho,\mathbf{u})\big) \nabla_\theta f(\mathbf{u};\theta)\mathrm{d}\mathbf{u}}{\int_{\mathscr{U}} S\big(J(\rho,\mathbf{u})\big) f(\mathbf{u};\theta)\mathrm{d}\mathbf{u}} \tag{9.15}$$

Now we apply the log trick $\nabla_\theta f(\mathbf{u};\theta) = f(\mathbf{u};\theta)\nabla_\theta \ln f(\mathbf{u};\theta)$ to obtain

$$\nabla_\theta \ln \int_{\mathscr{U}} S\big(J(\rho,\mathbf{u})\big) f(\mathbf{u};\theta)\mathrm{d}\mathbf{u} = \frac{\int_{\mathscr{U}} S\big(J(\rho,\mathbf{u})\big) f(\mathbf{u};\theta)\nabla_\theta \ln f(\mathbf{u};\theta)\mathrm{d}\mathbf{u}}{\int_{\mathscr{U}} S\big(J(\rho,\mathbf{u})\big) f(\mathbf{u};\theta)\mathrm{d}\mathbf{u}} \tag{9.16}$$

$$= \frac{\mathbb{E}_{f(\mathbf{u};\theta)}\Big[S\big(J(\rho,\mathbf{u})\big)\nabla_\rho \ln f(\mathbf{u};\theta)\Big]}{\mathbb{E}_{f(\mathbf{u};\theta)}\Big[S\big(J(\rho,\mathbf{u})\big)\Big]} \tag{9.17}$$

The exponential family distribution is given by

$$f(\mathbf{u};\theta) = h(\mathbf{u})\big(\theta^\top T(\mathbf{u}) - A(\theta)\big), \tag{9.18}$$

which is characterized by a set of natural parameters θ, sufficient statistics $T(\mathbf{u})$, base measure $h(\mathbf{u})$, and a log partition function $A(\theta)$. Thus we have

$$\nabla_\theta \ln f(\mathbf{u};\theta) = \nabla_\theta \ln \Big[h(\mathbf{u}) \exp\big(\theta^\top T(\mathbf{u}) - A(\theta)\big)\Big] \tag{9.19}$$

$$= \nabla_\theta \ln h(\mathbf{u}) + \nabla_\theta \big(\theta^\top T(\mathbf{u}) - A(\theta)\big) \tag{9.20}$$

$$= T(\mathbf{u}) - \nabla_\theta A(\theta) \tag{9.21}$$

If one optimizes only over the mean of a Gaussian distribution, then one obtains

$$\nabla_\theta \ln \mathbb{E}_{f(\mathbf{u};\theta)}\Big[S\big(J(\rho,\mathbf{u})\big)f(\mathbf{u};\theta)\Big] = \frac{\mathbb{E}_{f(\mathbf{u};\theta)}\Big[S\big(J(\rho,\mathbf{u})\big)\Sigma^{-1}\big(\mathbf{u}-\mu\big)\Big]}{\mathbb{E}_{f(\mathbf{u};\theta)}\Big[S\big(J(\rho,\mathbf{u})\big)\Big]}, \tag{9.22}$$

where μ is the mean, and Σ is the variance. Thus the gradient-ascent parameter update becomes

$$\Sigma^{-1}\mu^{k+1} = \Sigma^{-1}\mu^k + \Sigma^{-1}\frac{\mathbb{E}_{f(\mathbf{u};\theta)}\Big[S\big(J(\rho,\mathbf{u})\big)\big(\mathbf{u}-\mu\big)\Big]}{\mathbb{E}_{f(\mathbf{u};\theta)}\Big[S\big(J(\rho,\mathbf{u})\big)\Big]} \tag{9.23}$$

or more simply

$$\mu^{k+1} = \mu^k + \frac{\mathbb{E}_{f(\mathbf{u};\theta)}\Big[S\big(J(\rho,\mathbf{u})\big)\big(\mathbf{u}-\mu^k\big)\Big]}{\mathbb{E}_{f(\mathbf{u};\theta)}\Big[S\big(J(\rho,\mathbf{u})\big)\Big]} \tag{9.24}$$

Note that in the cases where we have added a level of abstraction due to the inclusion of a parameterized policy network $\Phi(\rho_t;\varphi)$, the above derivation yields a parameter update

$$\mu^{k+1} = \mu^k + \frac{\mathbb{E}_{f(\varphi;\theta)}\Big[S\big(J(\rho,\mathbf{u})\big)\big(\varphi-\mu^k\big)\Big]}{\mathbb{E}_{f(\varphi;\theta)}\Big[S\big(J(\rho,\mathbf{u})\big)\Big]}, \tag{9.25}$$

where in this case μ is the mean of a Gaussian distribution on the policy network parameters φ.

Due to the path integral nature of this derivation, the so called Quantum Gradient-based Adaptive Stochastic Search (QGASS) approach is independent of discretization scheme used to discretize the dynamics in eq. (9.1). Furthermore, this approach may consider discrete dynamics, such as in the discrete measurement case eq. (7.10).

Several of the above optimization problems contain two or three expectations, which is quite different than the above case wherein the parameter update was derived. In order to apply this parameter update to the two and three expectation cases above, one must simply

re-define the shape function. In the cases of eqs. (9.5), (9.7), (9.9) and (9.10), let the function $S(\cdot)$ be defined as

$$S(\cdot) := \mathbb{E}_Q\left[\hat{S}(\cdot)\right] \tag{9.26}$$

where $\hat{S}$ is a standard shape function which is differentiable, non-increasing, and positive semi-definite. Thus S is non-increasing and positive semi-definite, so it may be treated as a shape function. This shape function may be substituted into eq. (9.25) to yield

$$\mu^{k+1} = \mu^k + \frac{\mathbb{E}_{f(\varphi;\theta)}\left[\mathbb{E}_Q\left[\hat{S}\big(J(\rho,\mathbf{u})\big)\right](\varphi-\mu)\right]}{\mathbb{E}_{f(\varphi;\theta)}\left[\mathbb{E}_Q\left[\hat{S}\big(J(\rho,\mathbf{u})\big)\right]\right]}. \tag{9.27}$$

Similarly, for eq. (9.6), let the function $S(\cdot)$ be defined as

$$S(\cdot) := \mathbb{E}_Q\Big[\mathbb{E}_U\left[\hat{S}(\cdot)\right]\Big]. \tag{9.28}$$

In this case $S(\cdot)$ is also non-increasing and positive semi-definite, so it may be treated as a shape function. This results in the parameter update

$$\mu^{k+1} = \mu^k + \frac{\mathbb{E}_{f(\varphi;\theta)}\left[\mathbb{E}_Q\left[\mathbb{E}_U\left[\hat{S}\big(J(\rho,\mathbf{u})\big)\right]\right](\varphi-\mu)\right]}{\mathbb{E}_{f(\varphi;\theta)}\left[\mathbb{E}_Q\left[\mathbb{E}_U\left[\hat{S}\big(J(\rho,\mathbf{u})\big)\right]\right]\right]}. \tag{9.29}$$

The parameter update in eq. (9.25) can be connected to the information theoretic version of the Model Predictive Path Integral (MPPI) algorithm for classical systems [199], as first explored in [197]. The information theoretic MPPI algorithm applies an exponential shape function $S(y;\kappa) := \exp(-\kappa y)$ for $y, \kappa \in \mathbb{R}$, however other shape functions, such as the sigmoid function, are explored in [197].

The key difference between the QGASS approach compared to MPPI [199] is that MPPI

requires one to perform importance sampling, which presents challenges as discussed in chapter 8 above. Namely, the change of measures between the controlled and uncontrolled open quantum systems with QND measurement requires inversion of an operator that is singular in a multitude of realizable experiments, such as the two-qubit system and the homodyne system.

In the context of [64], policies without explicit time dependence have been shown to effectively control a number of SPDE systems for reaching and stabiliziation tasks, however these policies can fail for tracking tasks. Both of these approaches are algorithmically quite similar, and may have theoretic connections if one can connect the objective in GASS to an analogous free-energy relative entropy relationship. Aside from the differences in the resulting loss functional, another primary difference between the two approaches can be summarized by observing eq. (9.15), wherein one passes the gradient directly to the distribution $f(\varphi;\theta)$ and "skips" the implicit dependence of $S(J(\rho))$ on ρ. This "skipped" gradient path enables one to bypass the potential discontinuities and non-differentiability of J, however in some sense ignores these contributions to the total gradient. However the resulting algorithmic performance provides a strong outcome bias to ignore this "skipped" connection in favor of algorithmic flexibility.

9.2 QGASS Algorithm

The QGASS framework is a general framework for sampling based stochastic optimization for feedback control of open quantum systems. The use of stochastic approximation allows one to bypass all the inherent issues considered in chapter 8. Namely, this approach does not require importance sampling, and as such does not require a change of measures between open quantum systems. This approach bypasses discontinuities in the dynamics by applying stochastic approximation, so it can handle both continuous QND measurement and discrete QND measurement, as well as non-differentiable or discontinuous cost functions J. The path integral expectations also enable one to apply *any* discretization scheme to discretize

the quantum dynamics.

Implementation of the above framework requires that state trajectory data be collected from either simulated trajectories from a discretized version of eq. (9.1), or trajectory data collected from QND measurement of an open quantum system experiment. In the case of simulated trajectories, we emphasize that the algorithm is *agnostic* to the discretization scheme that is used to implement the approach in simulation on a computer.

The resulting algorithm is described in algorithm 6. Here we use the subscript notation $\rho_{t,r,p}$ to denote the density state ρ at time t for rollout r, and parameter realization p. This algorithm is specifically for the cases in eqs. (9.7), (9.9) and (9.10) which use deterministic parametric policies, however it is straightforward to extend algorithm 6 to the case of the stochastic policies in eq. (9.6). Similarly, one can reduce algorithm 6 to the case of parametric open loop policies in eq. (9.5).

The inputs to algorithm 6 may change depending on the specific problem (i.e. continuous QND measurement vs discrete QND measurement), however in most cases contain time horizon (T), number of iterations (K), number of rollouts (R), number of parameter realizations (P), initial state ρ_0, number of actuators (N), shape function parameter (κ), initial policy parameters ($\varphi^{(0)}$), initial distribution parameters ($\theta^{(0)}$), sample distribution variance (σ), and learning rate (γ). The *SampleWeights* method samples $f(\varphi;\theta)$ in the shape of the parameters φ. The *SampleNoise*() performs discrete samples of the Wiener process dW in eq. (9.1) as per the chosen discretization scheme. The *RunningCost* and *TerminalCost* methods are based on the typical splitting of the cost functional in eq. (7.15), wherein L denotes *RunningCost* and L_f denotes *TerminalCost*.

9.3 Simulated Results

The QGASS algorithm was implemented on the two qubit experiment detailed in section 8.1. The simulation environment was created in Python and utilizes some basic functionality of the QuTip [200] library, with policy networks coded using PyTorch [201]. The algorithm

Algorithm 6 Quantum Gradient-based Adaptive Stochastic Search Optimization

Function: $\Theta^* = \textbf{\textit{OptimizePolicyVars}}(T,K,R,P,\rho_0,N,\kappa,\varphi^{(0)},\theta^{(0)},\sigma,\gamma)$
for $k = 0$ to K **do**
  $\mu \leftarrow \theta^{(k)}$
  for $p = 0$ to P **do**
    $\varphi_p \leftarrow SampleWeights(\mu,\sigma)$
    for $r = 0$ to R **do**
      for $t = 0$ to T **do**
        $\mathrm{d}W_{t,r} \leftarrow SampleNoise()$
        $u_{t,r,p} \leftarrow Policy(\rho_{t,r,p};\varphi_p)$
        $\rho_{t+1,r,p} \leftarrow Propagate(\rho_{t,r,p},u_{t,r,p},\mathrm{d}W_{t,r})$ via eq. (9.1)
        $J_{t,r,p} \leftarrow RunningCost(\rho_{t,r,p},u_{t,r,p})$
      end for
      $J_{r,p} \leftarrow \sum_t J_{t,r,p} + TerminalCost(\rho_{T,r,p})$
    end for
    $S_p \leftarrow ShapeFunction(J_{r,p};\kappa)$
  end for
  $\theta^{(k+1)} \leftarrow \gamma GradientStep(\theta^{(k)},S_p)$ via eq. (9.27)
end for

computation speed is numerically improved by using vectorized (or batch) computations of the simulated trajectories, and CPU parellelization for policy parameter rollouts, resulting in $\tilde{2}0$ seconds per iteration for 1000 timesteps of an Euler-Maruyama discretization of eq. (9.1) with 50 rollouts and 200 policy parameter rollouts. The algorithm was run on a desktop computer with a Intel Xeon 12-core CPU with a NVIDIA GeForce GTX 1060 GPU, and used less than 10 GB of RAM.

The two qubit experiment involves a task wherein a random initial state must reach and stabilize to the symmetric maximally entangled Bell state eq. (8.15), restated here for clarity:

$$|\Psi^+\rangle = \frac{1}{\sqrt{2}}\Big(|\downarrow_1\uparrow_2\rangle + |\uparrow_1\downarrow_2\rangle\Big), \tag{9.30}$$

which can be written in density matrix form as

$$\rho_{\text{desired}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 \\ 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{9.31}$$

The initial state was sampled using the 'rand_ket' qutip method, which is then transformed into a normalized density matrix. This task used a single layer fully connected linear policy network which was initialized with the method in [202]. Performance of the state and control trajectories was measured by a running cost metric, given by

$$J(\rho, \mathbf{u}) := \int_0^T \Big(Q_s q\big(1 - \text{Tr}[\rho_{\text{desired}} \rho_\tau]\big) + \mathbf{u}_\tau^\top Q_u \mathbf{u}_\tau \Big) \mathrm{d}\tau, \tag{9.32}$$

where Q_s is a state cost weighting and Q_u is a control cost weighting. Note that this state cost metric utilizes a computationally efficient trace metric [195] as compared to the standard trace distance metric [200]

$$\text{Tracedist}(\rho_{\text{desired}}, \rho_t) = \frac{1}{2} \text{Re}\bigg(\sum_{i=1}^{n} \sqrt{\big| \lambda_i \big[(\rho_{\text{desired}}, -\rho_t)(\rho_{\text{desired}}, -\rho_t)^\dagger \big] \big|} \bigg), \tag{9.33}$$

which is substantially slower in implementation as it requires an eigenvalue decomposition at each time step. The function $q : [0,1] \to [0,\alpha]$ is an angle resolution function which is added to help resolve numerically "close" angular values. Recall that for a single qubit, the trace inner product can be thought of as measuring perpendicularity of Bloch phases. Since the cosine function is relatively flat (derivative near zero) near 0, one may encounter bad numerical resolution near the desired minimum $1 - \text{Tr}[\rho_{\text{desired}} \rho_\tau]\big) = 0$ in an n-qubit setting. The resolving function applies a logarithm transformation to improve numerical resolution,

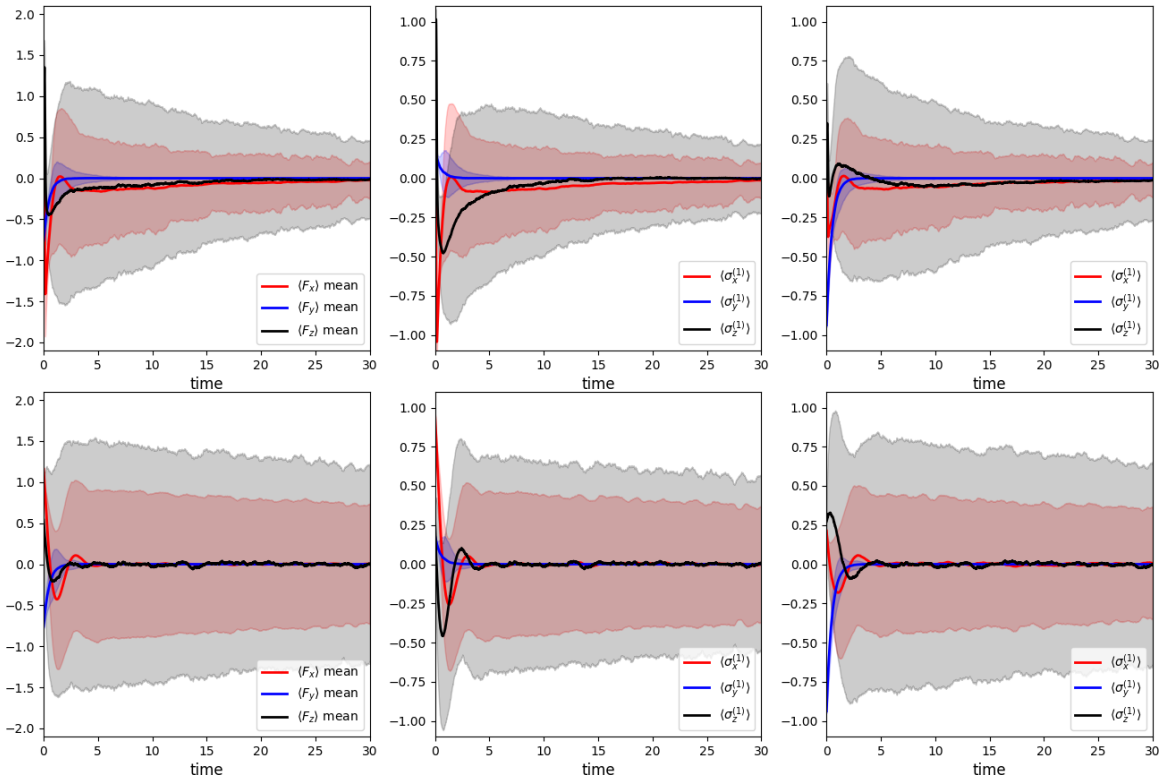


Figure 9.1: Two Qubit Symmetric Bell State Stabilization Task: (top) Control with a linear policy trained by the QGASS algorithm, (bottom) Control with the policy suggested by Mirrahimi, et. al. Colored lines denote the mean trajectories of the two qubit basis elements, and shaded regions denote 2-σ variances. Means and variances are taken over 1000 trajectory rollouts.

and is given by

$$q(x) = \alpha \frac{\log(1+\beta x)}{\log(1+\beta)}, \tag{9.34}$$

where α is the maximum of the range, and β controls the slope by effectively changing the base of the natural logarithm.

The linear policy was trained using 850 iterations of QGASS over a time window of 10 seconds, with 200 parameter rollouts per iteration and 50 rollouts of the dynamics per parameter rollout. Despite being trained on just 10 seconds of dynamics, the linear policy was tested on up to 1000 seconds of dynamics and remained performant over the entire test window. The combined degrees of freedom and individual qubit degrees of freedom of the QGASS trained linear policy and the policy presented in Mirrahimi, et. al. [195] are shown

in fig. 9.1. The left subplot depicts the expectation of the state with respect to the F_x, F_y, and F_z operators given by

$$F_j = \sigma_j^{(1)} + \sigma_j^{(2)}, \quad j \in \{x, y, z\} \tag{9.35}$$

$$= \sigma_j \otimes I + I \otimes \sigma_j, \quad j \in \{x, y, z\}. \tag{9.36}$$

As usual, expectations of operators are computed as

$$\langle A \rangle := \mathrm{Tr}\big(A\rho\big). \tag{9.37}$$

Depicted in the center subplot are the first qubit degrees of freedom, and in the right subplot the second qubit degrees of freedom. Each subplot has a solid line for the mean and shading for the 2-σ variance as computed over 1000 rollouts of the open system dynamics. These results depict a smaller 2-σ variance over the controlled trajectory produced by the policy trained by QGASS as compared to the controlled trajectory produced by the policy suggested by Mirrahimi, et. al.

While this visualization is consistent with QuTip documentation [200], it does not capture the full behavior of the system. Recall that the space of two qubit density matrices is spanned by the set of combinations of direct product operators of the single qubit density basis $\{I, \sigma_x, \sigma_y, \sigma_z\}$, i.e. two qubit span $= \{I \otimes \sigma_x, I \otimes \sigma_y, I \otimes \sigma_z, \sigma_x \otimes I, \sigma_x \otimes \sigma_x, \sigma_x \otimes \sigma_y, \ \ldots\}$, which has 16 elements. In this basis, the desired density matrix can be represented as a

linear combination of basis elements as

$$\rho_{\text{desired}} = \frac{1}{4}\left(I \otimes I + \sigma_x \otimes \sigma_x + \sigma_y \otimes \sigma_y - \sigma_z \otimes \sigma_z\right) \tag{9.38}$$

$$= \frac{1}{4}\left(\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\right) \tag{9.39}$$

Thus, achieving the desired state can be viewed more directly through expectations of these basis elements, as depicted in fig. 9.2. In this representation, the convergence behavior is clear. The time horizon in fig. 9.2 is longer as compared to fig. 9.1, which is required to visualize the asymptotic convergence of the pertinent basis elements. Thus one can also observe that the original single qubit spin expectations were obfuscating the true system behavior.

The top subfigure depicts the QGASS method, and the bottom subfigure depicts the method suggested by Mirrahimi, et. al. [195]. Each solid line in each subfigure represents the mean expectation of the basis element, averaged over 1000 system rollouts, and the shading represents the 2-σ variance of the distribution of expectations. In this basis, the QGASS method can be observed to converge in approximately one order of magnitude faster than the benchmark, and has dramatically lower variance than the benchmark.

The efficacy of the policy trained by QGASS can also be visualized in terms of the cost metric in eq. (9.32). In fig. 9.3, the running cost components of the policy trained by QGASS are depicted. The left subfigure depicts the running state cost component of eq. (9.32), given

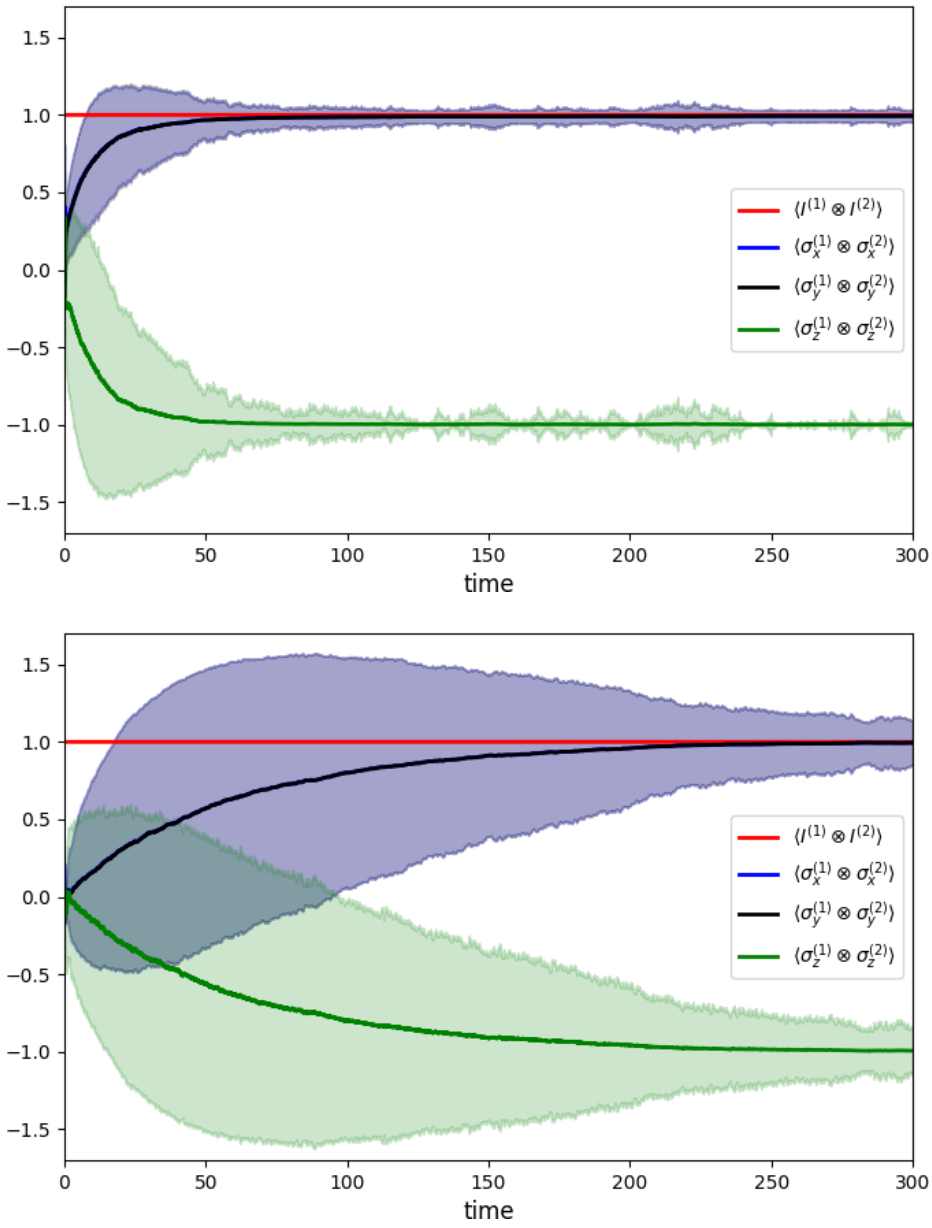


Figure 9.2: Two Qubit Symmetric Bell State Stabilization Task: (top) Control with a linear policy trained by the QGASS algorithm, (bottom) Control with the policy suggested by Mirrahimi, et. al. Colored lines denote the mean trajectories of the two qubit basis elements, and shaded regions denote 2-σ variances. Means and variances are taken over 1000 trajectory rollouts.

by

$$J_{\text{state},t}(\rho,t) := \int_0^t Q_s q\big(1 - \text{Tr}[\rho_{\text{desired}}\rho_\tau]\big)\,\mathrm{d}\tau, \tag{9.40}$$

while the right subfigure depicts the running control cost component of eq. (9.32), given by

$$J_{\text{control},t}(\mathbf{u}) := \int_0^t \mathbf{u}_\tau^\top Q_u \mathbf{u}_\tau \mathrm{d}\tau. \tag{9.41}$$

The solid line depicts the means of the running cost trajectories of each policy and the shading depicts the 1-σ variance, each computed over 1000 trajectory rollouts. The policy trained by QGASS has a lower state cost on average, with a significantly lower 1-σ variance of state cost, which suggests that the state performance, as measured by the running state cost metric, may have better guarantees of performance as compared to the policy suggested by Mirrahimi, et. al. The control effort of each policy is depicted in the right subfigure. It can be observed that the policy trained by QGASS applies a strong initial control impulse to the system, followed by a relatively small control signal. This policy can be interpreted as a form of “bang-bang" control. In contrast, the policy suggested by Mirrahimi, et. al. injects a fairly constant control signal over the time window, which yields a cumulative control effort which is approximately 12 times higher than the policy trained by QGASS.

Note that the impulsive control signal produced by the trained policy is likely to be experimentally realizable due to the viability of pulsed electromagnetic fields, which are used in a variety of scientific and non-scientific applications. However, if one were to desire a less impulsive control signal, one could add a running penalization term on the derivative of the control, effectively penalizing large rates of change in the control signal applied to the system [147]. One could also add a control rate indicator, effectively suppressing this additional cost until some control rate threshold is reached. This flexibility is possible since we do *not* require any differentiability or even continuity of the cost functional in the QGASS framework. While this will likely lead to a larger time-integrated control effort, one may introduce terms to the cost functional which aid in meeting various experimental hardware constraints, including bounds on the control rate.

These results demonstrate efficacy of the QGASS framework for the two-qubit experi-

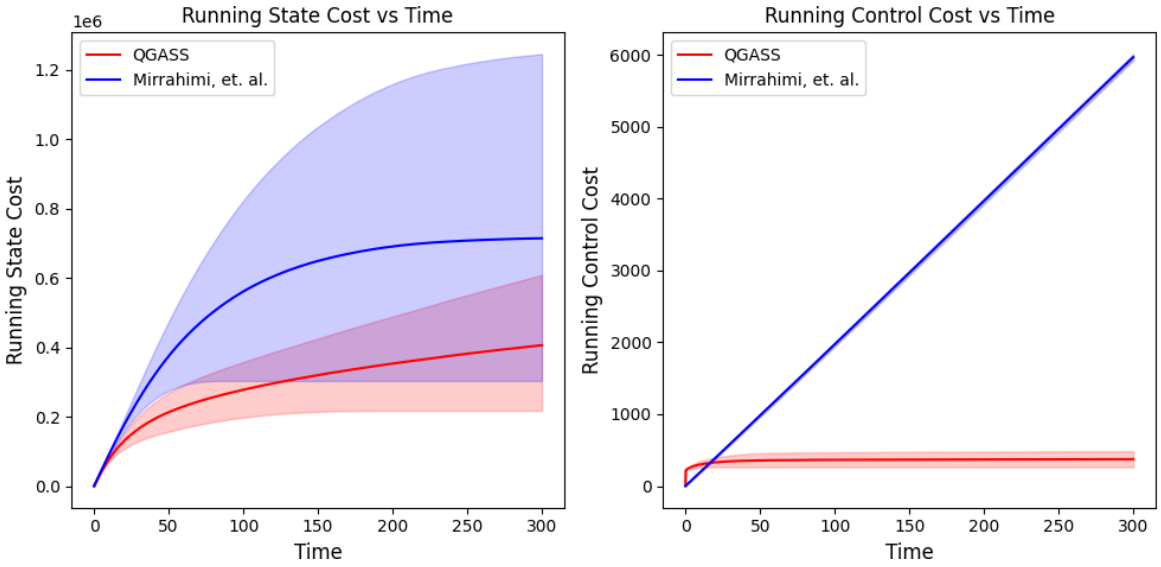


Figure 9.3: Two Qubit Symmetric Bell State Stabilization Task: (left) running state costs and (right) running control costs for the linear policy network trained with QGASS and the policy suggested by Mirrahimi, et. al. Colored lines denote the means and shaded regions denote 1-σ variances. Means and variances are taken over 1000 trajectory rollouts. The imposed data symmetry from this Gaussian depiction is corrected only when a large variance would lead to an infeasible negative instaneous cost.

ment, however the generality of the problem formulation suggests that this approach can be applied to the large class of experiments involving open quantum systems with either continuous QND measurement schemes or discrete QND measurement schemes. Furthermore, this approach is quite flexible; while we have only considered one form of cost functional and shape functional, there are virtually no limitations on the form of the cost functional, and there is quite a large variety of possible shape functionals. Combined with the fast computational speed of iterations, the results suggest that this approach can scale to larger numbers of qubits, which is actively being explored.

Also worth exploring is the size and depth of the policy network. The above results utilized a policy network that is rather shallow and simple. The widespread success of utilizing deep networks for a variety of learning applications suggests that deeper network architectures may outperform the results presented here, especially for experiments with larger numbers of qubits.

9.4 Conclusion

In this chapter, we develop a general optimization approach for feedback control of open quantum systems with QND measurement. We propose several feedback control models, and derive a parameter update scheme which can be used in each case to optimize the parameters of the sampling distribution. The resulting QGASS algorithm is applied to the two qubit experiment, and the results demonstrate significant outperformance of a related feedback control approach.

The results presented here are encouraging. In future work, the author plans to scale the approach to larger qubit systems, test several of the variants suggested in this chapter, and apply second-order optimization methods in the GASS literature. The author also plans to extend the current framework to the context of optimal gate synthesis.

CHAPTER 10

VARIATIONAL OPTIMIZATION FOR SAMPLING-BASED DYNAMIC COMPENSATION OF SMES

In this section, we revisit the information theoretic variational optimization approach taken in chapter 8. Therein, a closed-loop feedback control architecture was derived based on the Legendre transformation, the so called free energy relative entropy relationship. The KL distance was used to penalize the 'distance' to the optimal measure resulting from minimization of the Legendre transformation. The change of measures, or RN derivative was applied twice in order to 1) change the sampling distribution from the optimal measure, which cannot be directly sampled, and 2) perform importance sampling so that each iteration can bootstrap off of the improvement from the previous iteration through biased sampling. The fundamental problem with that approach is that the change of measures between the uncontrolled and controlled measures of the open quantum system dynamics with QND measurement yields inversion of the covariance of the noise term $\mathrm{d}W_t$, which in many realistic experiments becomes singular and thus an inverse does not exist.

As a result of this realization, in chapter 9 a formulation was derived which is simpler and yields strong results in simulation, yet is no longer directly connected with the foundational information theoretic relationship. Not only is the free energy relative entropy relation an instantiation of the second law of thermodynamics, but it has demonstrated connections to foundational notions in SOC literature, namely the HJB equation.

In this section, we revisit the information theoretic perspective, and introduce an augmented dynamic system which can overcome the inversion difficulties introduced by the change of measures. Consider again the simplified form of the open quantum system

dynamics subject to a continuous QND measurement process

$$\mathrm{d}\rho_t = F(\rho_t)\mathrm{d}t + G(\rho_t, \mathbf{u}_t)\mathrm{d}t + B(\rho_t)\mathrm{d}W_t, \tag{10.1}$$

where $B(\rho_t)$ is singular in many cases. Without loss of generality, assume these dynamics evolve a vectorized density matrix. Now, let the control $\mathbf{u}$ be generated by a process

$$\mathrm{d}\mathbf{u}_t = R(\rho_t)\Big(L(t,\rho_t)\mathrm{d}t + \frac{1}{\sqrt{\zeta}}\mathrm{d}V_t\Big), \tag{10.2}$$

where $L(t,\rho_t)$ is some user-defined function, $\mathrm{d}V_t$ is a standard (classical) zero-mean Wiener process, $R(\rho_t)$ is a positive-definite covariance operator, and $\zeta \in \mathbb{R}$. Now, we wish to find an "uncontrolled" analog of this controller, which is the result of turning off the function $L(t,\rho_t)$, given by

$$\mathrm{d}\mathbf{u}_t = \frac{1}{\sqrt{\zeta}}R(\rho_t)\mathrm{d}V_t. \tag{10.3}$$

We will refer to eq. (10.3) as the "unforced" dynamic compensator and eq. (10.2) as the "forced" dynamic compensator. With abuse of notation, if we augment eq. (9.1) with eq. (10.2), we get the following augmented dynamics

$$\mathrm{d}\hat{\rho}_t := \mathrm{d}\begin{bmatrix} \rho_t \\ \mathbf{u}_t \end{bmatrix} \tag{10.4}$$

$$= \begin{bmatrix} F(\rho_t) + G(\rho_t, \mathbf{u}_t) \\ 0 \end{bmatrix}\mathrm{d}t + \begin{bmatrix} 0 \\ R(\rho_t)L(t,\rho_t) \end{bmatrix}\mathrm{d}t + \begin{bmatrix} B(\rho_t) & 0 \\ 0 & \frac{1}{\sqrt{\zeta}}R(\rho_t) \end{bmatrix}\begin{bmatrix} \mathrm{d}W_t \\ \mathrm{d}V_t \end{bmatrix}. \tag{10.5}$$

It is important to note here that the Wiener process $\mathrm{d}W_t$ does not enter the compensator dynamics, however the Wiener process $\mathrm{d}V_t$ *does* enter the dynamics of ρ_t through second-order effects. Thus the stochastic process ρ_t has two sources of stochasticity. The decoupling

of $\mathrm{d}W_t$ from the compensator dynamics allows us to write eq. (10.5) in the following form

$$\mathrm{d}\hat{\rho}_t = \begin{bmatrix} F(\rho_t) + G(\rho_t, \mathbf{u}_t) \\ 0 \end{bmatrix} \mathrm{d}t + \begin{bmatrix} B(\rho_t) & 0 \\ 0 & \frac{1}{\sqrt{\zeta}} R(\rho_t) \end{bmatrix} \left(\begin{bmatrix} 0 \\ \sqrt{\zeta} L(t, \rho_t) \end{bmatrix} \mathrm{d}t + \begin{bmatrix} \mathrm{d}W_t \\ \mathrm{d}V_t \end{bmatrix} \right) \tag{10.6}$$

$$= \hat{F}(\hat{\rho}_t)\mathrm{d}t + \hat{B}(\rho_t)\Big(\hat{L}(\rho_t)\mathrm{d}t + \mathrm{d}\hat{W}_t\Big), \tag{10.7}$$

where $\hat{F}(\cdot)$, and $\hat{B}(\cdot)$, $\hat{L}(\cdot)$ are defined according to the associated augmented operator functions, and $\hat{W}_t$ is the augmented Wiener process, which is just a standard (classical) Wiener process.

Note that in general $\rho \in \mathscr{H}$, however $\mathbf{u}$ can have different dimensionality depending on the problem scope. In order to generalize, we define the real Hilbert space $\mathscr{H}_U$ with inner product $\langle \cdot, \cdot \rangle_{\mathscr{H}_U}$ and say $u \in \mathscr{H}_U$, where the bold face script is dropped since it is now a Hilbert space vector as opposed to a finite vector. Thus we can think of $\hat{\rho}$ as belonging to an orthogonal direct sum Hilbert space $\hat{\rho} \in \mathscr{H}_A := \mathscr{H} \oplus \mathscr{H}_U$, with inner product $\langle \cdot, \cdot \rangle$. Note also that due to the form of $\hat{F}(\cdot)$, the augmented dynamics form a *semilinear* SPDE, with Cylindrical Wiener process $\hat{W} \in \mathscr{H}_{\hat{W}}$ defined by orthogonal direct sum $\mathscr{H}_{\hat{W}} := L_2(\mathbb{R}) \oplus \mathscr{H}_U$.

Theorem 10.1 (Girsanov)**.** *Let $\hat{W}_t \in \mathscr{H}_{\hat{W}}$ be a standard Wiener process on $(\Omega, \mathscr{F}, \mathscr{F}_t, \mathbb{P})$. Consider the unforced and forced nonlinear stochastic processes,*

$$\mathrm{d}\hat{\rho}(t) = \hat{F}(\hat{\rho})\mathrm{d}t + \hat{B}(\hat{\rho})\mathrm{d}\hat{W}(t) \tag{10.8}$$

$$\mathrm{d}\tilde{\hat{\rho}}(t) = \hat{F}(\tilde{\hat{\rho}})\mathrm{d}t + \hat{B}(\tilde{\hat{\rho}})\Big(\hat{L}(t, \tilde{\hat{\rho}})\mathrm{d}t + \mathrm{d}\hat{W}(t)\Big). \tag{10.9}$$

Assume the stochastic processes are well posed and have unique $\mathscr{F}_t$-adapted solutions $\hat{\rho}(t)$ and $\tilde{\hat{\rho}}(t)$, $t \geq 0$. Let Γ be a set of continuous time trajectories in the interval $[0, T]$. Define the probability law of $\hat{\rho}(t)$ over trajectories Γ as $\mathscr{L}(\Gamma) := \mathbb{P}\big(\omega \in \Omega \mid \hat{\rho}(\cdot, \omega) \in \Gamma\big)$ and

similarly define the law of $\tilde{\hat{\rho}}(t)$ *as* $\tilde{\mathscr{L}}(\Gamma) := \mathbb{P}\big(\omega \in \Omega \mid \tilde{\hat{\rho}}(\cdot,\omega) \in \Gamma\big)$. *Define*

$$\varphi(t) := \hat{L}\big(t, \hat{\rho}(t)\big) \tag{10.10}$$

and assume

$$\mathbb{E}_{\mathbb{P}}\left[\exp\Big(\frac{1}{2}\int_0^T ||\varphi(t)||^2 \mathrm{d}s\Big)\right] < +\infty \tag{10.11}$$

Then

$$\tilde{\mathscr{L}}(\Gamma) = \mathbb{E}_{\mathbb{P}}\left[\exp\Big(\int_0^T \Big\langle \varphi(s), \mathrm{d}\hat{W}(s)\Big\rangle - \frac{1}{2}\int_0^T ||\varphi(s)||^2 \mathrm{d}s\Big)\,\Big|\,\hat{\rho}(\cdot) \in \Gamma\right] \tag{10.12}$$

Proof. This proof will take many identical steps to theorem C.1, which is similarly proven for SPDEs in Hilbert spaces. First, define the process

$$\tilde{\hat{W}}(t) := \hat{W}(t) - \int_0^t \varphi(s)\mathrm{d}s. \tag{10.13}$$

Under the assumption in eq. (10.11), and applying [2, Theorem 10.14], $\tilde{\hat{W}}$ is a Wiener process with respect to a measure $\mathbb{Q}$ defined by

$$\mathrm{d}\mathbb{Q}(\omega) := \exp\left(\int_0^t \Big\langle \varphi(s), \mathrm{d}\hat{W}(s)\Big\rangle - \frac{1}{2}\int_0^T ||\varphi(s)||^2 \mathrm{d}s\right)\mathrm{d}\mathbb{P} \tag{10.14}$$

$$= \exp\left(\int_0^t \Big\langle \varphi(s), \mathrm{d}\tilde{\hat{W}}(s)\Big\rangle + \frac{1}{2}\int_0^T ||\varphi(s)||^2 \mathrm{d}s\right)\mathrm{d}\mathbb{P} \tag{10.15}$$

Next, we use eq. (10.13) to rewrite eq. (10.8) with forcing as

$$\mathrm{d}\hat{\rho}(t) = \hat{F}(\hat{\rho})\mathrm{d}t + \hat{B}(\hat{\rho})\mathrm{d}\hat{W}(t) \tag{10.16}$$

$$= \hat{F}(\hat{\rho})\mathrm{d}t + \hat{B}(\hat{\rho})\left(\hat{L}(t,\hat{\rho})\mathrm{d}t + \mathrm{d}\tilde{\hat{W}}(t)\right). \tag{10.17}$$

Notice that the SPDE in eq. (10.17) has the same form as eq. (10.9). Therefore under measure $\mathbb{Q}$ and Wiener process $\tilde{\hat{W}}(t)$, $\tilde{\hat{\rho}}(\cdot,\omega)$ is equivalent to $\hat{\rho}(\cdot,\omega)$. Furthermore, under

measure $\mathbb{P}$, eq. (10.8) and eq. (10.17) also describe the same system on $(\Omega, \mathscr{F}, \mathscr{F}_t, \mathbb{P})$. Thus, from the assumption of uniqueness of solutions, one has

$$\mathbb{P}\big(\{\tilde{\hat{\rho}} \in \Gamma\}\big) = \mathbb{Q}\big(\{\hat{\rho} \in \Gamma\}\big). \tag{10.18}$$

Thus we have obtained the result, which is given by eq. (10.14). □

Now, applying the Girsanov theorem theorem 10.1 with the fact that for any function $\lambda \in C([0,T]; \mathscr{H}_A)$ one has [2, Chapter 1]

$$\mathbb{E}_{\mathbb{P}}[\lambda(Z)] = \int_{\Omega} \lambda\big(Z(\cdot,\omega)\big)\mathrm{d}\mathbb{P}(\omega) = \int_{C([0,T];H)} \lambda(\mathsf{x})\mathrm{d}\mathscr{L}(\mathsf{x}), \quad \mathsf{x} \in \Gamma, \tag{10.19}$$

we obtain the change of measures, or RN derivative,

$$\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}} = \exp\bigg(-\int_0^T \Big\langle \varphi(s), \mathrm{d}\tilde{\tilde{W}}(s) \Big\rangle - \frac{1}{2}\int_0^T ||\varphi(s)||^2 \mathrm{d}s \bigg) \tag{10.20}$$

$$= \exp\bigg(-\int_0^T \Big\langle \hat{L}\big(s, \hat{\rho}(s)\big), \mathrm{d}\tilde{\tilde{W}}(s) \Big\rangle - \frac{1}{2}\int_0^T ||\hat{L}\big(s, \hat{\rho}(s)\big)||^2 \mathrm{d}s \bigg). \tag{10.21}$$

Next we apply the property of orthogonal sum Hilbert spaces. Let H_1, H_2 be two Hilbert spaces, and let $H_{\text{sum}} = H_1 \oplus H_2$ be an orthogonal direct sum Hilbert space. Then one has the following

$$\langle a, b\rangle_{H_{\text{sum}}} = \langle a_1, b_1\rangle_{H_1} + \langle a_2, b_2\rangle_{H_2}, \quad a, b \in H_{\text{sum}},\ a = a_1 \oplus a_2,\ b = b_1 \oplus b_2. \tag{10.22}$$

Thus the change of measures, or RN derivative becomes

$$\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}} = \exp\bigg(-\int_0^T \Big\langle 0, \mathrm{d}W(s) \Big\rangle_{L_2(\mathbb{R})} - \sqrt{\zeta}\int_0^T \Big\langle L\big(\rho(s)\big), \mathrm{d}V(s) \Big\rangle_{\mathscr{H}_U} \tag{10.23}$$

$$+ \frac{1}{2}\int_0^T 0\mathrm{d}s + \frac{\zeta}{2}\int_0^T ||L\big(s, \rho(s)\big)||^2_{\mathscr{H}_U}\mathrm{d}s \bigg) \tag{10.24}$$

$$= \exp\bigg(-\sqrt{\zeta}\int_0^T \Big\langle L\big(\rho(s)\big), \mathrm{d}V(s) \Big\rangle_{\mathscr{H}_U} + \frac{\zeta}{2}\int_0^T ||L\big(s, \rho(s)\big)||^2_{\mathscr{H}_U}\mathrm{d}s \bigg) \tag{10.25}$$

Now, define the process

$$\tilde{V}(t) := V(t) - \sqrt{\zeta}\int_0^t L\big(s,\rho(s)\big)\mathrm{d}s \tag{10.26}$$

and note that $\tilde{V}(t) \in \mathscr{H}_U$ is also a Wiener process via [2, Theorem 10.14]. This yields the change of measures

$$\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}} = \exp\bigg(-\sqrt{\zeta}\int_0^T \Big\langle L\big(s,\rho(s)\big), \mathrm{d}\tilde{V}(s)\Big\rangle_{\mathscr{H}_U} - \frac{\zeta}{2}\int_0^T ||L\big(s,\rho(s)\big)||^2_{\mathscr{H}_U}\mathrm{d}s\bigg). \tag{10.27}$$

The most critical feature of the change of measures in this form is that it does *not* require any inversion. In fact, it is straightforward to work out that the change of measures eq. (10.27) is *equivalent* to a change of measures between properly defined measures on the forced and unforced dynamic compensator dynamics in eqs. (10.2) and (10.3). This will allow us to proceed to derive open loop, MPC, and explicit feedback policies derived for open quantum systems analogous to those derived in previous chapters for classical systems.

This is achieved by applying the methodology in chapter 4. For some abstract state cost functional $J = J(\hat{\rho})$, we write the free energy-relative entropy with constant $r \in \mathbb{R}$ as

$$-\frac{1}{r}\log\mathbb{E}_{\mathscr{L}}\Big[\exp(-rJ)\Big] = \min_{L(t,\rho)}\Big[\mathbb{E}_{\tilde{\mathscr{L}}}(J) + \frac{1}{r}D_{KL}(\tilde{\mathscr{L}}\,\big|\big|\,\mathscr{L})\Big], \tag{10.28}$$

which has the Gibbs minimizing measure

$$\tilde{\mathscr{L}}^* = \frac{\exp(-rJ)\mathrm{d}\mathscr{L}}{\mathbb{E}_{\mathscr{L}}\big[\exp(-rJ)\big]} \tag{10.29}$$

From here, one can derive various control methods depending on the parameterization of the dynamic compensator forcing function $L(t,\rho_t)$.

10.1 Variational Optimization for Open Loop and MPC Quantum Dynamic Compensator Policies

This approach is analogous to the open loop and MPC approach developed in chapter 4, and will be referred to as Quantum Variational Optimization - Single Shot (QVO-SS) and Quantum Variational Optimization - MPC (QVO-MPC), respectively. We begin by dropping the explicit state dependence of the dynamic compensator forcing function

$$L(t,\rho_t) = L(t) \tag{10.30}$$

Despite the focus of this chapter to derive feedback policies for open quantum systems, in this section we arrive at an open loop iterative update scheme. As will become apparent, the resulting iterative scheme still incorporates state information *implicitly*, and furthermore can be applied in a MPC setting, which more closely resembles an explicit feedback control architecture.

With this parameterization, the optimization problem takes the form

$$L^*(t) = \underset{L}{\operatorname{argmin}} D_{KL}\left(\tilde{\mathscr{L}}^* || \tilde{\mathscr{L}}\right) \tag{10.31}$$

$$= \underset{L}{\operatorname{argmin}} \int_\Omega \log\left(\frac{\mathrm{d}\tilde{\mathscr{L}}^*}{\mathrm{d}\tilde{\mathscr{L}}}\right) \mathrm{d}\tilde{\mathscr{L}}^* \tag{10.32}$$

$$= \underset{L}{\operatorname{argmin}} \int_\Omega \log\left(\frac{\mathrm{d}\tilde{\mathscr{L}}^*}{\mathrm{d}\mathscr{L}}\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}}\right) \mathrm{d}\tilde{\mathscr{L}}^* \tag{10.33}$$

which is equivalent to minimizing

$$L^*(t) = \underset{L}{\operatorname{argmin}} \int_\Omega \log\left(\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}}\right) \mathrm{d}\tilde{\mathscr{L}}^* \tag{10.34}$$

$$= \underset{L}{\operatorname{argmin}} \mathbb{E}_{\tilde{\mathscr{L}}^*}\left[-\sqrt{\zeta}\int_0^T \left\langle L(s), \mathrm{d}V(s)\right\rangle_{\mathscr{H}_U} + \frac{\zeta}{2}\int_0^T ||L(s)||^2_{\mathscr{H}_U}\mathrm{d}s\right] \tag{10.35}$$

In light of a discrete time implementation, it suffices to consider the class of step functions

L_i, $i = 0, \ldots, N_T - 1$ that are constant over fixed-size intervals $[t_i, t_{i+1}]$ of length Δt

$$L^*(t) = \underset{L}{\operatorname{argmin}} \left(-\sqrt{\zeta} \sum_{i=0}^{N_T-1} \mathbb{E}_{\mathscr{L}^*} \left[\int_{t_i}^{t_{i+1}} \left\langle L_i, \mathrm{d}V(s) \right\rangle_{\mathscr{H}_U} \right] + \frac{\zeta}{2} \sum_{i=0}^{N_T-1} \left\langle L_i, L_i \right\rangle_{\mathscr{H}_U} \Delta t \right) \tag{10.36}$$

Now, assume that

$$\mathbb{E}_{\mathscr{L}^*} \left[\int_{t_i}^{t_{i+1}} \left| \left\langle L_i, \mathrm{d}V(s) \right\rangle \right| \right] < +\infty, \quad \forall i = 0, 1, 2, \ldots. \tag{10.37}$$

Then, under a properly formulated Fubini's Theorem, one has

$$L^*(t) = \underset{L}{\operatorname{argmin}} \left(-\sqrt{\zeta} \sum_{i=0}^{N_T-1} \left\langle L_i, \mathbb{E}_{\mathscr{L}^*} \left[\int_{t_i}^{t_{i+1}} \mathrm{d}V(s) \right] \right\rangle_{\mathscr{H}_U} + \frac{\zeta}{2} \sum_{i=0}^{N_T-1} \left\langle L_i, L_i \right\rangle_{\mathscr{H}_U} \Delta t \right), \tag{10.38}$$

which, by taking partial derivative $\frac{\partial}{\partial L_i}$ and setting equal to zero (i.e. performing a Newton step), yields the optimal forcing

$$L_i^* = \frac{1}{\sqrt{\zeta}\Delta t} \mathbb{E}_{\mathscr{L}^*} \left[\int_{t_i}^{t_{i+1}} \mathrm{d}V(s) \right]. \tag{10.39}$$

Since we cannot sample directly from $\tilde{\mathscr{L}}^*$, one must express the above expectation with respect to the measure of the controlled dynamics $\tilde{\mathscr{L}}$. This step is referred to as importance

sampling, and yields an iterative approach, where iterations are denoted by k

$$L_i^{(k+1)} = \frac{1}{\sqrt{\zeta}\Delta t}\int_\Omega \int_{t_i}^{t_{i+1}} \mathrm{d}V(s)\mathrm{d}\tilde{\mathscr{L}}^* \tag{10.40}$$

$$= \frac{1}{\sqrt{\zeta}\Delta t}\int_\Omega \frac{\mathrm{d}\tilde{\mathscr{L}}^*}{\mathrm{d}\mathscr{L}}\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}}\int_{t_i}^{t_{i+1}} \mathrm{d}V(s)\mathrm{d}\tilde{\mathscr{L}} \tag{10.41}$$

$$= \frac{1}{\sqrt{\zeta}\Delta t}\int_\Omega \frac{\exp(-rJ)}{\mathbb{E}_{\mathscr{L}}\left[\exp(-rJ)\right]}\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}}\int_{t_i}^{t_{i+1}} \mathrm{d}V(s)\mathrm{d}\tilde{\mathscr{L}} \tag{10.42}$$

$$= \frac{1}{\sqrt{\zeta}\Delta t}\int_\Omega \frac{\exp(-rJ)}{\mathbb{E}_{\tilde{\mathscr{L}}}\left[\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}}\exp(-rJ)\right]}\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}}\int_{t_i}^{t_{i+1}} \mathrm{d}V(s)\mathrm{d}\tilde{\mathscr{L}} \tag{10.43}$$

$$= \frac{1}{\sqrt{\zeta}\Delta t}\mathbb{E}_{\tilde{\mathscr{L}}}\left[\frac{\exp(-rJ^{(k)})}{\mathbb{E}_{\tilde{\mathscr{L}}}\left[\exp(-rJ^{(k)})\right]}\int_{t_i}^{t_{i+1}} \mathrm{d}V(s)\right] \tag{10.44}$$

where $J^{(k)}$ is introduced as a result of importance sampling, and is given by

$$J^{(k)}(\hat{\rho}) := J(\hat{\rho}) + \frac{\sqrt{\zeta}}{r}\sum_{i=0}^{N_T-1}\left\langle L_i^{(k)}, \int_{t_i}^{t_{i+1}} \mathrm{d}\tilde{V}(s)\right\rangle_{\mathscr{H}_U} + \frac{\zeta\Delta t}{2r}\sum_{i=0}^{N_T-1}\left\langle L_i^{(k)}, L_i^{(k)}\right\rangle_{\mathscr{H}_U} \tag{10.45}$$

Finally, rewriting eq. (10.26) as

$$V(t) = \tilde{V}(t) + \sqrt{\zeta}\Delta t L_i^{(k)}, \tag{10.46}$$

one has the iterative forcing update scheme

$$L_i^{(k+1)} = L_i^{(k)} + \frac{1}{\sqrt{\zeta}\Delta t}\mathbb{E}_{\tilde{\mathscr{L}}}\left[\frac{\exp(-rJ^{(k)})}{\mathbb{E}_{\tilde{\mathscr{L}}}\left[\exp(-rJ^{(k)})\right]}\int_{t_i}^{t_{i+1}} \mathrm{d}\tilde{V}(s)\right]. \tag{10.47}$$

One can easily note the similarities between this forcing update scheme and the control update scheme in eq. (4.16), especially in the subcase $r = \zeta$. In many ways, our ability to derive such an approach is akin to the QGASS approach, wherein the optimization problem is made independent of the difficulties inherent in the original optimization problem. In this case it is achieved by controlling the forcing of the dynamic compensator, while in the case of QGASS it is achieved by optimizing the shape of the sampling distribution.

While appearing to be completely open loop, and completely independent of the original physical system, this approach directly incorporates state information through the cost functional $J^{(k)}$. This approach can be thought of as an *implicit* feedback scheme, which still requires continuous state measurements provided by the QND measurement scheme. Furthermore, as seen in chapter 4, such a scheme can also be applied in an 'online' MPC setting. Here, one optimizes the trajectory of the forcing function $L(t)$ on a finite subinterval $[t_{\text{sim}}, T_{\text{sim}}] \subset [0,T]$ for K iterations, applies control at the current time step, and then recedes the subinterval backward by a time step Δt. The process is repeated until $T_{\text{sim}} = T$. Algorithmic details of the MPC approach can be found in algorithm 3. In the quantum case, the algorithm is otherwise identical except for trading the classical SPDE dynamics for the augmented quantum dynamics eq. (10.9).

10.2 Variational Optimization for Learning Dynamic Compensator Policies with Explicit Feedback

In light of open loop and MPC dynamic compensator policies in section 10.1, which is derived in analogous approach to chapter 4, the approach in this section is analogous to the explicit feedback approach developed in chapters 5 and 6. Consider the forcing function L with an explicit state dependence,

$$L(t,\rho_t) = L(t,\hat{\rho}_t;\Theta), \tag{10.48}$$

where Θ are a finite set of parameters. Note also that the dependence on $\hat{\rho}$ enables one to incorporate density information in addition to the internal dynamic compensator state. The variational optimization problem becomes

$$\Theta^* = \underset{\Theta}{\operatorname{argmin}}\, D_{KL}\big(\tilde{\mathscr{L}}^* || \tilde{\mathscr{L}}\big). \tag{10.49}$$

Expanding the KL divergence and applying the chain rule yields

$$\Theta^* = \underset{\Theta}{\operatorname{argmin}} \left[\int_\Omega \log\Big(\frac{\mathrm{d}\tilde{\mathscr{L}}^*}{\mathrm{d}\mathscr{L}} \frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}} \Big) \mathrm{d}\tilde{\mathscr{L}}^* \right] \tag{10.50}$$

$$= \underset{\Theta}{\operatorname{argmin}} \left[\int_\Omega \Big(\log \frac{\mathrm{d}\tilde{\mathscr{L}}^*}{\mathrm{d}\mathscr{L}} + \log \frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}} \Big) \mathrm{d}\tilde{\mathscr{L}}^* \right], \tag{10.51}$$

which due to a lack of dependence of $\frac{\mathrm{d}\tilde{\mathscr{L}}^*}{\mathrm{d}\mathscr{L}}$ on Θ, is equivalent to minimizing

$$\Theta^* = \underset{\Theta}{\operatorname{argmin}} \left[\int_\Omega \log\Big(\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}} \Big) \mathrm{d}\tilde{\mathscr{L}}^* \right]. \tag{10.52}$$

Once again, since one cannot directly sample from $\tilde{\mathscr{L}}^*$, we perform importance sampling in order to sample instead from the controlled measure $\tilde{\mathscr{L}}$. This yields

$$\Theta^* = \underset{\Theta}{\operatorname{argmin}} \left[\int_\Omega \log\Big(\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}} \Big) \frac{\mathrm{d}\tilde{\mathscr{L}}^*}{\mathrm{d}\mathscr{L}} \frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}} \mathrm{d}\tilde{\mathscr{L}} \right]. \tag{10.53}$$

Plugging eq. (10.29) and eq. (10.27) into eq. (10.53) yields

$$\Theta^* = \underset{\Theta}{\operatorname{argmin}} \left[\int_\Omega \frac{\exp(-rJ)}{\mathbb{E}_{\mathscr{L}}\big[\exp(-rJ)\big]} \frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}} \mathrm{d}\tilde{\mathscr{L}} \Big(- \sqrt{\zeta} \int_0^T \Big\langle L\big(s, \hat{\rho}(s); \Theta\big), \mathrm{d}\tilde{V}(s) \Big\rangle_{\mathscr{H}_U} - \frac{\zeta}{2} \int_0^T ||L\big(s, \hat{\rho}(s); \Theta\big)||^2_{\mathscr{H}_U} \mathrm{d}s \Big) \right] \tag{10.54}$$

$$= \underset{\Theta}{\operatorname{argmin}} \left[\int_\Omega \frac{\exp(-rJ)}{\mathbb{E}_{\tilde{\mathscr{L}}}\big[\exp(-rJ)\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}}\big]} \frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}} \mathrm{d}\tilde{\mathscr{L}} \Big(- \sqrt{\zeta} \int_0^T \Big\langle L\big(s, \hat{\rho}(s); \Theta\big), \mathrm{d}\tilde{V}(s) \Big\rangle_{\mathscr{H}_U} - \frac{\zeta}{2} \int_0^T ||L\big(s, \hat{\rho}(s); \Theta\big)||^2_{\mathscr{H}_U} \mathrm{d}s \Big) \right] \tag{10.55}$$

$$= \underset{\Theta}{\operatorname{argmin}} \mathbb{E}_{\tilde{\mathscr{L}}} \left[\frac{\exp(-r\tilde{J})}{\mathbb{E}_{\tilde{\mathscr{L}}}\big[\exp(-r\tilde{J})\big]} \Big(- \sqrt{\zeta} \int_0^T \Big\langle L\big(s, \hat{\rho}(s); \Theta\big), \mathrm{d}\tilde{V}(s) \Big\rangle_{\mathscr{H}_U} - \frac{\zeta}{2} \int_0^T ||L\big(s, \hat{\rho}(s); \Theta\big)||^2_{\mathscr{H}_U} \mathrm{d}s \Big) \right], \tag{10.56}$$

where $\tilde{J} = \tilde{J}\big(\hat{\rho}, \Theta\big)$ is the importance sampled cost defined as

$$\tilde{J}\big(\hat{\rho}, \Theta\big) := J(\hat{\rho}) + \frac{\sqrt{\zeta}}{r} \int_0^T \Big\langle L\big(s, \hat{\rho}(s); \Theta\big), \mathrm{d}\tilde{V}(s) \Big\rangle_{\mathscr{H}_U} + \frac{\zeta}{2r} \int_0^T ||L\big(s, \hat{\rho}(s); \Theta\big)||^2_{\mathscr{H}_U} \mathrm{d}s \tag{10.57}$$

For simplicity, let us introduce the following functions

$$\mathscr{N}(\Theta) := \int_0^T \Big\langle L\big(s, \hat{\rho}(s); \Theta\big), \mathrm{d}\tilde{V}(s) \Big\rangle_{\mathscr{H}_U}, \tag{10.58}$$

$$\mathscr{P}(\Theta) := \int_0^T ||L\big(s, \hat{\rho}(s); \Theta\big)||^2_{\mathscr{H}_U}, \tag{10.59}$$

$$\mathscr{E}(\hat{\rho}, \Theta) := \frac{\exp(-r\tilde{J})}{\mathbb{E}_{\tilde{\mathscr{L}}}\big[\exp(-r\tilde{J})\big]}. \tag{10.60}$$

Thus our optimization problem takes a simplified form

$$\Theta^* = \underset{\Theta}{\operatorname{argmin}}\, \mathbb{E}_{\tilde{\mathscr{L}}} \left[\mathscr{E}(\hat{\rho}, \Theta) \left(-\sqrt{\zeta}\mathscr{N}(\Theta) - \frac{\zeta}{2}\mathscr{P}(\Theta) \right) \right] \tag{10.61}$$

$$\tilde{J}\big(\hat{\rho}, \Theta\big) = J(\hat{\rho}) + \frac{\sqrt{\zeta}}{r}\mathscr{N}(\Theta) + \frac{\zeta}{2r}\mathscr{P}(\Theta) \tag{10.62}$$

This problem formulation is extremely flexible; one can use a variety of specific forms of $L(t, \hat{\rho}_t; \Theta)$, one can apply a variety of methods to perform minimization, and one can apply this approach to virtually *any* open quantum system with QND measurement. Here, we will consider forcing functions which are neural policy networks that take the state ρ as input, and output the forcing. Such a policy network could be the network architectures used in chapter 5, i.e. a Fully Connected (FC) network or a CNN, however one may also consider the compensator dynamics as a neural SDE [203, 204] of particular forms.

Neural networks of all such forms have had widespread success in deep learning applications, which often apply some variant of gradient descent (e.g. SGD or ADAM [109, 110]) in order to perform the minimization in eq. (10.61). In such a context, one may consider the

loss function

$$\mathfrak{L}(\Theta) := \mathbb{E}_{\tilde{\mathscr{L}}}\left[\mathscr{E}(\hat{\rho},\Theta)\left(-\sqrt{\zeta}\mathscr{N}(\Theta) - \frac{\zeta}{2}\mathscr{P}(\Theta)\right)\right]. \tag{10.63}$$

Applying a gradient based approach leads to an update of the form

$$\Theta^{(k+1)} = \Theta^{(k)} - \gamma_{\Theta}\nabla_{\Theta}\mathfrak{L}(\Theta^{(k)}) \tag{10.64}$$

where ∇_{Θ} denotes the Gateaux partial derivative with respect to Θ, γ_{Θ} is a learning rate, and the superscript k denotes iteration. The resulting approach is called Quantum Spatio-Temporal Stochastic Optimization (QSTSO), and is algorithmically identical to algorithm 4 except for trading classical dynamics for the augmented quantum dynamics in eq. (10.9), and omitting the $\mathbf{m}(\mathbf{x})$ and $\mathbf{M}(\mathbf{x})$ computation.

In each of these three sampling based control optimization methods for open quantum systems, the difficulties inherent in the dynamics of the system are avoided by a form of abstraction; rather than deal directly with the control policy, in QGASS we optimize over the parameters of the distribution from which we sample control policies, and in QVO-SS, QVO-MPC, and QSTSO we optimize over the forcing function of the dynamic compensator. These methods have been shown to be effective in the context of spatio-temporal systems, and show promise in the context of open quantum systems with QND feedback.

10.3 Conclusion

This chapter revisits the ITC variational optimization based approach taken in chapter 8, and augments the open quantum system dynamics with a stochastic dynamic compensator driven by a standard Wiener process. We prove a version of Girsanov's theorem for the augmented quantum system, and arrive at a change of measures, or RN derivative, that critically does not involve the inversion of an operator. In section 10.1, the inversion-free change of measures is used to develop implicit feedback control approaches akin to those derived in chapter 4. Namely we develop the QVO-SS and the QVO-MPC approaches to

iteratively optimize the forcing function of the dynamic compensator. In section 10.2, the inversion-free change of measures is used to develop an approach akin to the approach in chapter 5. The resulting SGD-based method, called QSTSO, can be used to optimize the loss function in eq. (10.63) with respect to the parameters of the forcing function of the dynamic compensator.

Based on the results in chapter 4, chapter 5, and chapter 6, the two approaches in this chapter are appealing. They offer generality in the types of systems that they can apply to, and they offer flexibility in the form of the varieties of control that can be applied. In future work, these methods will be implemented for two qubit systems, and compared to the QGASS framework. The authors will also scale these approaches to larger systems.

CHAPTER 11
CONCLUSION

This thesis develops numerous optimization approaches for control of spatio-temporal systems. These range from forward-backward schemes as in chapter 3, to sampling-based second-order optimization schemes in chapters 4 and 8 and section 10.1, to gradient-based optimization schemes in chapters 5, 6, 9 and 10. These approaches are developed from a perspective that unifies stochastic optimal control theory and information theoretic control theory, and are developed from a set of unifying mathematics that allow treatment of both macroscopic and microscopic systems. Such algorithms are applied to the optimal control problems and optimal control and co-design problems in fluid dynamics, robotics, and quantum mechanics.

This thesis opens doors to control and co-design applications in a multitude of diverse systems that were not explored here, from magnetic confinement fusion reactors, to complex weather systems, to morphing wings, and many more. Beyond exploring diverse applications of this work, other future work includes exploring scalability to 3D systems and larger scale 2D systems, which in many cases may leverage recent neural architectures and computational architectures that leverage sparsity. In the quantum regime, a major future effort is dedicated to applying the dynamic compensator approaches, scaling to larger qubit systems, exploring discrete photon counting experiments, and exploring applications in optimal universal quantum gate synthesis.

Appendices

APPENDIX A

DESCRIPTION OF THE HILBERT SPACE WIENER PROCESS

In this section we provide formal definitions of various forms of the Hilbert space Wiener process. Some of these statements can be found in [2, Section 4.1].

Definition A.1. *Let $\mathscr{H}$ denote a Hilbert space. A $\mathscr{H}$-valued stochastic process $W(t)$ with probability law $\mathscr{L}\big(W(\cdot)\big)$ is called a Wiener process if*

i) $W(0) = 0$

ii) *W has continuous trajectories*

iii) *W has independent increments*

iv) $\mathscr{L}\big(W(t) - W(s)\big) = \mathscr{N}\big(0, (t-s)Q\big), \quad t \geq s \geq 0$

v) $\mathscr{L}\big(W(t)\big) = \mathscr{L}\big(-W(t)\big), \quad t \geq 0$

Proposition A.1. *Let $\{e_i\}_{i=1}^{\infty}$ be a complete orthonormal system for the Hilbert Space $\mathscr{H}$. Let Q denote the covariance operator of the Wiener process $W(t)$. Note that Q satisfies $Qe_i = \lambda_i e_i$, where λ_i is the eigenvalue of Q that corresponds to eigenvector e_i. Then, $W(t) \in \mathscr{H}$ has the following expansion:*

$$W(t) = \sum_{j=1}^{\infty} \sqrt{\lambda_j}\beta_j(t)e_j, \tag{A.1}$$

where $\beta_j(t)$ are real valued Brownian motions that are mutually independent on $(\Omega, \mathscr{F}, \mathbb{P})$.

Definition A.2. *Let $\{e_i\}_{i=1}^{\infty}$ be a complete orthonormal system for the Hilbert Space $\mathscr{H}$. An operator A on $\mathscr{H}$ with the set of its eigenvalues $\{\lambda_i\}_{i=1}^{\infty}$ in a given basis $\{e_i\}_{i=1}^{\infty}$ is called*

a trace-class operator if

$$Tr(A) := \sum_{n=1}^{\infty} \langle Ae_n, e_n \rangle = \sum_{i=1}^{\infty} \lambda_i < \infty. \tag{A.2}$$

The two primary Wiener processes that are typically used to model spatio-temporal noise processes in the SPDE literature are the Cylindrical Wiener process and the Q-Wiener process. These are both referred to in the main text, and are defined in the following two definitions.

Definition A.3. *A Wiener process $W(t)$ on $\mathscr{H}$ with covariance operator Q is called a Cylindrical Wiener process if Q is the identity operator I.*

Definition A.4. *A Wiener process $W(t)$ on $\mathscr{H}$ with covariance operator Q is called a Q-Wiener process if Q is of trace-class.*

An immediate fact following definition A.3 is that the Cylindrical Wiener process acts spatially *everywhere* on $\mathscr{H}$ with equal magnitude. One can easily conclude that for a Cylindrial Wiener process, the eigenvalues $\{\lambda_i\}_{i=1}^{\infty}$ of the covariance operator Q are all unity, thus

$$\sum_{i=1}^{\infty} \lambda_i = \infty. \tag{A.3}$$

However, we note that in this case the series in eq. (A.1) converges in another Hilbert space $U_1 \supset U$, when the inclusion $\iota : U \to U_1$ is Hilbert-Schmidt. For more details see [70].

On the other hand, immediately following definition A.4 is the fact that a Q-Wiener process must not have a spatially equal effect everywhere on the domain. More precisely, one has the following proposition.

Proposition A.2. *Let $W(t)$ be a Q-Wiener process on $\mathscr{H}$ with covariance operator Q. Let $\{\lambda_i\}_{i=1}^{\infty}$ denote the set of eigenvalues of Q in the complete orthonormal system $\{e_i\}_{i=1}^{\infty}$ such that $\sum_i \lambda_i < \infty$. Then the eigenvalues must fall into one of the following three cases:*

i) For any $\varepsilon > 0$, there are only finitely many eigenvalues λ_i of covariance operator Q such

that $|\lambda_i| > \varepsilon$. *That is, the set* $\{i \in \mathbb{N}_+ : |\lambda_i| > \varepsilon\}$, *where* $\mathbb{N}_+$ *is the positive natural numbers, has finite elements.*

ii) The eigenvalues λ_i *of covariance operator Q follow a bounded periodic function such that* $|\lambda_i| > 0 \; \forall \; i \in \mathbb{N}_+$ *and* $\sum_{i=1}^{\infty} \lambda_i = 0$.

iii) Both case i) and case ii) are satisfied. In this case the eigenvalues follow a bounded and convergent periodic function with $\lim_{i \to \infty} \lambda_i = 0$.

APPENDIX B

RELATIVE ENTROPY AND FREE ENERGY DUALITIES IN HILBERT SPACES

In this section we provide the relation between free energy and relative entropy. This connection is valid for general probability measures, including measures defined on path spaces induced by infinite-dimensional stochastic systems. In what follows, L^p $(1 \leq p < \infty)$ denotes the standard L^p space of measurable functions and $\mathscr{P}$ denotes the set of probability measures.

Definition B.1. *(Free Energy) Let $\mathscr{L} \in \mathscr{P}$ a probability measure on a sample space Ω, and consider a measurable function $J : L^p \to \mathbb{R}_+$. Then the following term:*

$$V := \frac{1}{\rho} \log_e \int_\Omega \exp(\rho J) \mathrm{d}\mathscr{L}(\omega), \tag{B.1}$$

is called the free energy[1] *of J with respect to $\mathscr{L}$ and $\rho \in \mathbb{R}$.*

Definition B.2. *(Generalized Entropy) Let $\mathscr{L}, \tilde{\mathscr{L}} \in \mathscr{P}$, then the relative entropy of $\tilde{\mathscr{L}}$ with respect to $\mathscr{L}$ is defined as:*

$$S\left(\tilde{\mathscr{L}} || \mathscr{L}\right) := \begin{cases} -\int_\Omega \frac{\mathrm{d}\tilde{\mathscr{L}}(\omega)}{\mathrm{d}\mathscr{L}(\omega)} \log_e \frac{\mathrm{d}\tilde{\mathscr{L}}(\omega)}{\mathrm{d}\mathscr{L}(\omega)} \mathrm{d}\mathscr{L}(\omega), \text{if } \tilde{\mathscr{L}} << \mathscr{L}, \\ +\infty, \quad \text{otherwise}, \end{cases}$$

where "<<" denotes absolute continuity of $\tilde{\mathscr{L}}$ with respect to $\mathscr{L}$. We say that $\tilde{\mathscr{L}}$ is absolutely continuous with respect to $\mathscr{L}$ and we write $\tilde{\mathscr{L}} << \mathscr{L}$ if $\mathscr{L}(B) = 0 \Rightarrow \tilde{\mathscr{L}}(B) = 0$, $\forall B \in \mathscr{F}$.

The free energy and relative entropy relationship is expressed by the following theorem:

[1] The function $\log_e$ denotes the natural logarithm.

Theorem B.1. *Let $(\Omega, \mathscr{F})$ be a measurable space. Consider $\mathscr{L}, \tilde{\mathscr{L}} \in \mathscr{P}$ and definitions B.1, B.2. Under the assumption that $\tilde{\mathscr{L}} << \mathscr{L}$, the following inequality holds:*

$$-\frac{1}{\rho}\log_e \mathbb{E}_{\mathscr{L}}\left[\exp(-\rho J)\right] \leq \left[\mathbb{E}_{\tilde{\mathscr{L}}}(J) - \frac{1}{\rho}S\left(\tilde{\mathscr{L}}\,||\,\mathscr{L}\right)\right], \tag{B.2}$$

where $\mathbb{E}_{\mathscr{L}}, \mathbb{E}_{\tilde{\mathscr{L}}}$ denote expectations under probability measures $\mathscr{L}$, $\tilde{\mathscr{L}}$ respectively. Moreover, $\rho \in \mathbb{R}_+$ and $J : L^p \to \mathbb{R}_+$. The inequality in eq. (B.2) *is the so called Legendre Transform.*

By defining the free energy as temperature $T = \frac{1}{\rho}$, the Legendre transformation has the form:

$$V \leq E - TS, \tag{B.3}$$

and the equilibrium probability measure has the classical form:

$$\mathrm{d}\mathscr{L}^*(\omega) = \frac{\exp(-\rho J)\mathrm{d}\mathscr{L}(\omega)}{\int_\Omega \exp(-\rho J)\mathrm{d}\mathscr{L}(\omega)}, \tag{B.4}$$

To verify the optimality of $\mathscr{L}^*$, it suffices to substitute eq. (B.4) in eq. (B.2) and show that the inequality collapses to an equality [99]. The statistical physics interpretation of inequality eq. (B.3) is that, maximization of entropy results in reduction of the available energy. At the thermodynamic equilibrium the entropy reaches its maximum and $V = E - TS$.

APPENDIX C

A GIRSANOV THEOREM FOR SPDES WITH CYLINDRICAL WIENER NOISE

Theorem C.1 (Girsanov)**.** *Let Ω be a sample space with a σ-algebra $\mathscr{F}$. Consider the following H-valued stochastic processes:*

$$\mathrm{d}X = \big(\mathscr{A}X + F(t,X)\big)\mathrm{d}t + G(t,X)\mathrm{d}W(t), \tag{C.1}$$

$$\mathrm{d}\tilde{X} = \big(\mathscr{A}\tilde{X} + F(t,\tilde{X})\big)\mathrm{d}t + \tilde{B}(t,\tilde{X})\mathrm{d}t + G(t,\tilde{X})\mathrm{d}W(t), \tag{C.2}$$

where $X(0) = \tilde{X}(0) = x$ and $W \in U$ is a cylindrical Wiener process with respect to measure $\mathbb{P}$. Moreover, for each $\Gamma \in C([0,T];H)$, let the law of X be defined as $\mathscr{L}(\Gamma) := \mathbb{P}(\omega \in \Omega | X(\cdot,\omega) \in \Gamma)$. Similarly, the law of $\tilde{X}$ is defined as $\tilde{\mathscr{L}}(\Gamma) := \mathbb{P}(\omega \in \Omega | \tilde{X}(\cdot,\omega) \in \Gamma)$. Assume

$$\mathbb{E}_{\mathbb{P}}\big[e^{\frac{1}{2}\int_0^T ||\psi(t)||^2 \mathrm{d}t}\big] < +\infty, \tag{C.3}$$

where

$$\psi(t) := G^{-1}\big(t,X(t)\big)\tilde{B}\big(t,X(t)\big) \in U_0. \tag{C.4}$$

Then

$$\tilde{\mathscr{L}}(\Gamma) = \mathbb{E}_{\mathbb{P}}\big[\exp\big(\int_0^T \langle \psi(s), dW(s)\rangle_U - \tfrac{1}{2}\int_0^T ||\psi(s)||_U^2 ds\big) | X(\cdot) \in \Gamma\big]. \tag{C.5}$$

Proof. Define the process:

$$\hat{W}(t) := W(t) - \int_0^t \psi(s)\mathrm{d}s. \tag{C.6}$$

Under the assumption in eq. (C.3), $\hat{W}$ is a cylindrical Wiener process with respect to a

measure $\mathbb{Q}$ determined by:

$$\begin{aligned} \mathrm{d}\mathbb{Q}(\omega) &= \exp\Big(\int_0^T \langle \psi(s), \mathrm{d}W(s)\rangle_U - \frac{1}{2}\int_0^T ||\psi(s)||_U^2 \mathrm{d}s\Big)\mathrm{d}\mathbb{P} \\ &= \exp\Big(\int_0^T \langle \psi(s), \mathrm{d}\hat{W}(s)\rangle_U + \frac{1}{2}\int_0^T ||\psi(s)||_U^2 \mathrm{d}s\Big)\mathrm{d}\mathbb{P}. \end{aligned} \tag{C.7}$$

The proof for this result can be found in [70, Theorem 10.14]. Now, using eq. (C.6), eq. (C.1) will be rewritten as:

$$\mathrm{d}X = \big(\mathscr{A}X + F(t,X)\big)\mathrm{d}t + G(t,X)\mathrm{d}W(t) \tag{C.8}$$

$$= \big(\mathscr{A}X + F(t,X)\big)\mathrm{d}t + B(t,X)\mathrm{d}t + G(t,X)\mathrm{d}\hat{W}(t) \tag{C.9}$$

Notice that the SPDE in eq. (C.9) has the same form as eq. (C.2). Therefore, under the introduced measure $\mathbb{Q}$ and noise profile $\hat{W}$, $X(\cdot,\omega)$ becomes equivalent to $\tilde{X}(\cdot,\omega)$ from eq. (C.2). Conversely, under measure $\mathbb{P}$, eq. (C.8) (or eq. (C.9)) behaves as the original system in eq. (C.1). In other words, eq. (C.1) and eq. (C.9) describe the same system on $(\Omega,\mathscr{F},\mathbb{P})$. From the uniqueness of solutions and the aforementioned reasoning, one has:

$$\mathbb{P}\big(\{\tilde{X}\in\Gamma\}\big) = \mathbb{Q}\big(\{X\in\Gamma\}\big).$$

The result follows from eq. (C.7). □

APPENDIX D

PROOF OF LEMMA 4.1

Proof. Under the open loop parameterization $\mathscr{U}(\mathbf{x},t) = \mathbf{m}(\mathbf{x})^{\mathrm{T}}\mathbf{u}(t)$, the problem takes the form:

$$\mathbf{u}^* = \operatorname{argmin}\left[\int_C \log_e \frac{\mathrm{d}\mathscr{L}^*(\mathsf{x})}{\mathrm{d}\tilde{\mathscr{L}}(\mathsf{x})}\mathrm{d}\mathscr{L}^*(\mathsf{x})\right] = \operatorname{argmin}\left[\int_C \log_e \frac{\mathrm{d}\mathscr{L}^*(\mathsf{x})}{\mathrm{d}\mathscr{L}(\mathsf{x})}\frac{\mathrm{d}\mathscr{L}(\mathsf{x})}{\mathrm{d}\tilde{\mathscr{L}}(\mathsf{x})}\mathrm{d}\mathscr{L}^*(\mathsf{x})\right].$$

By using the change of measures in eq. (4.12) of the main text, minimization of the last expression is equivalent to the minimization of the expression:

$$\mathbb{E}_{\mathscr{L}^*}\left[\log_e \frac{\mathrm{d}\mathscr{L}(\mathsf{x})}{\mathrm{d}\tilde{\mathscr{L}}(\mathsf{x})}\right] = -\sqrt{\rho}\mathbb{E}_{\mathscr{L}^*}\left[\int_0^T \mathbf{u}(t)^\top \bar{\mathbf{m}}(t)\right] + \frac{1}{2}\rho\mathbb{E}_{\mathscr{L}^*}\left[\int_0^T \mathbf{u}(t)^\top \mathbf{M}\mathbf{u}(t)\mathrm{d}t\right].$$

As stated in lemma 4.1, we apply the control in discrete time instances, and consider the class of step functions $\mathbf{u}_i$, $i = 0,\ldots,L-1$ that are constant over fixed-size intervals $[t_i, t_{i+1}]$ of length Δt:

$$\mathbb{E}_{\mathscr{L}^*}\left[\log_e \frac{\mathrm{d}\mathscr{L}(\mathsf{x})}{\mathrm{d}\tilde{\mathscr{L}}(\mathsf{x})}\right] = -\sqrt{\rho}\sum_{i=0}^{L-1}\mathbf{u}_i^\top \mathbb{E}_{\mathscr{L}^*}\left[\int_{t_i}^{t_{i+1}} \bar{\mathbf{m}}(t)\right] + \frac{1}{2}\rho\sum_{i=0}^{L-1}\mathbf{u}_i^\top \mathbf{M}\mathbf{u}_i\Delta t,$$

where we have used the fact that $\mathbf{M}$ is constant with respect to time. Due to the symmetry of $\mathbf{M}$, minimization of the expression above with respect to $\mathbf{u}_i$ results in:

$$\mathbf{u}_i^* = \frac{1}{\sqrt{\rho}\Delta t}\mathbf{M}^{-1}\mathbb{E}_{\mathscr{L}^*}\left[\int_{t_i}^{t_{i+1}} \bar{\mathbf{m}}(t)\right]. \tag{D.1}$$

Since we cannot sample directly from the optimal measure $\mathscr{L}^*$, we need to express the above expectation with respect to the measure induced by controlled dynamics, $\mathscr{L}^{(i)}$. We can then directly sample controlled trajectories based on $\mathscr{L}^{(i)}$ and approximate the optimal

control trajectory. The change in expectation is achieved by applying the Radon-Nikodym derivative. These so called importance sampling steps are as follows. First define $W^{(i)}$ in a similar fashion to eq. (C.6), as:

$$W^{(i)}(t) := W(t) - \int_0^t \sqrt{\rho}\mathscr{U}^{(i)}(s)\mathrm{d}s. \tag{D.2}$$

Similar to eq. (C.9), one can rewrite the uncontrolled dynamics as

$$\begin{aligned} \mathrm{d}X &= \big(\mathscr{A}X + F(t,X)\big)\mathrm{d}t + \frac{1}{\sqrt{\rho}}G(t,X)\mathrm{d}W(t) \\ &= \big(\mathscr{A}X + F(t,X)\big)\mathrm{d}t + G(t,X)\big(\mathscr{U}^{(i)}\mathrm{d}t + \frac{1}{\sqrt{\rho}}\mathrm{d}W^{(i)}(t)\big). \end{aligned} \tag{D.3}$$

Under the open loop parameterization $\mathscr{U}(\mathbf{x},t) = \mathbf{m}(\mathbf{x})^\top \mathbf{u}_j$, where $\mathbf{u}_j$ are step functions on each interval $[t_j, t_{j+1}]$, the change of measures becomes

$$\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\mathscr{L}^{(i)}} = \exp\bigg(-\sqrt{\rho}\sum_{k=0}^{L-1} \mathbf{u}_k^{(i)\top} \int_{t_k}^{t_{k+1}} \bar{\mathbf{m}}^{(i)}(t) - \rho\frac{1}{2}\sum_{k=0}^{L-1} \mathbf{u}_k^{(i)\top}\mathbf{M}\mathbf{u}_k^{(i)}\Delta t\bigg), \tag{D.4}$$

where

$$\bar{\mathbf{m}}^{(i)}(t) := \Big[\langle m_1, \mathrm{d}W^{(i)}(t)\rangle_U, ..., \langle m_N, \mathrm{d}W^{(i)}(t)\rangle_U\Big]^\top \in \mathbb{R}^N. \tag{D.5}$$

One can alternatively write this as

$$\begin{aligned} \bigg(\int_{t_j}^{t_{j+1}} \bar{\mathbf{m}}^{(i)}(t)\bigg)_l &= \int_{t_j}^{t_{j+1}} \big\langle m_l, \mathrm{d}W^{(i)}(t)\big\rangle_U = \int_{t_j}^{t_{j+1}} \big\langle m_l, \mathrm{d}W(t) - \sqrt{\rho}\mathscr{U}^{(i)}(t)\mathrm{d}t\big\rangle_U \\ &= \int_{t_j}^{t_{j+1}} \big\langle m_l, \mathrm{d}W(t)\big\rangle_U - \sqrt{\rho}\Big[\big\langle m_l, m_1\big\rangle_U, ..., \big\langle m_l, m_N\big\rangle_U\Big]\mathbf{u}_j^{(i)}\Delta t. \end{aligned}$$

It follows that:

$$\int_{t_j}^{t_{j+1}} \bar{\mathbf{m}}^{(i)}(t) = \int_{t_j}^{t_{j+1}} \bar{\mathbf{m}}(t) - \sqrt{\rho}\Delta t\mathbf{M}\mathbf{u}_j^{(i)}. \tag{D.6}$$

In order to arrive at the iterative scheme, we perform one step of importance sampling

and express the associated expectations with respect the measure induced by the controlled SPDE in eq. (4.2) of the main text. Let us begin by modifying eq. (D.1) via the appropriate change of measures from eq. (D.4), as well as eq. (D.2):

$$\begin{aligned}
\mathbf{u}_j^{i+1} &= \frac{1}{\sqrt{\rho}\Delta t}\mathbf{M}^{-1}\int_\Omega \left[\frac{\mathrm{d}\mathscr{L}^*}{\mathrm{d}\mathscr{L}}\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\mathscr{L}^{(i)}}\int_{t_i}^{t_{i+1}}\bar{\mathbf{m}}(t)\right]\mathrm{d}\mathscr{L}^{(i)} && \text{(D.7)}\\
&= \frac{1}{\sqrt{\rho}\Delta t}\mathbf{M}^{-1}\int\left[\frac{\exp(-\rho J)}{\mathbb{E}_{\mathscr{L}}\left[\exp(-\rho J)\right]}\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\mathscr{L}^{(i)}}\int_{t_i}^{t_{i+1}}\bar{\mathbf{m}}(t)\right]\mathrm{d}\mathscr{L}^{(i)}\\
&= \frac{1}{\sqrt{\rho}\Delta t}\mathbf{M}^{-1}\int\left[\frac{\exp(-\rho J)}{\mathbb{E}_{\mathscr{L}^{(i)}}\left[\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\mathscr{L}^{(i)}}\exp(-\rho J)\right]}\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\mathscr{L}^{(i)}}\int_{t_i}^{t_{i+1}}\bar{\mathbf{m}}(t)\right]\mathrm{d}\mathscr{L}^{(i)}\\
&= \frac{1}{\sqrt{\rho}\Delta t}\mathbf{M}^{-1}\mathbb{E}_{\mathscr{L}^{(i)}}\left[\frac{\exp(-\rho J^{(i)})}{\mathbb{E}_{\mathscr{L}^{(i)}}\left[\exp(-\rho J^{(i)})\right]}\int_{t_i}^{t_{i+1}}\bar{\mathbf{m}}(t)\right]. && \text{(D.8)}
\end{aligned}$$

One can reorder eq. (D.6) as

$$\int_{t_j}^{t_{j+1}}\bar{\mathbf{m}}(t) = \int_{t_j}^{t_{j+1}}\bar{\mathbf{m}}^{(i)}(t) + \sqrt{\rho}\Delta t\mathbf{M}\mathbf{u}_j^{(i)}, \tag{D.9}$$

and plug it into eq. (D.8) to yield:

$$\begin{aligned}
\mathbf{u}_j^{i+1} &= \frac{1}{\sqrt{\rho}\Delta t}\mathbf{M}^{-1}\mathbb{E}_{\mathscr{L}^{(i)}}\left[\frac{\exp(-\rho J^{(i)})}{\mathbb{E}_{\mathscr{L}^{(i)}}\left[\exp(-\rho J^{(i)})\right]}\int_{t_j}^{t_{j+1}}\bar{\mathbf{m}}^{(i)}(t) + \sqrt{\rho}\Delta t\mathbf{M}\mathbf{u}_j^{(i)}\right]\\
&= \mathbf{u}_j^{(i)} + \frac{1}{\sqrt{\rho}\Delta t}\mathbf{M}^{-1}\mathbb{E}_{\mathscr{L}^{(i)}}\left[\frac{\exp(-\rho J^{(i)})}{\mathbb{E}_{\mathscr{L}^{(i)}}\left[\exp(-\rho J^{(i)})\right]}\int_{t_j}^{t_{j+1}}\bar{\mathbf{m}}^{(i)}(t)\right], && \text{(D.10)}
\end{aligned}$$

which is equivalent to eq. (4.16) in the main text with $J^{(i)}$ defined by eq. (4.17) in the main text. □

APPENDIX E

FEYNMAN-KAC FOR SPATIO-TEMPORAL DIFFUSIONS: FROM EXPECTATIONS TO HILBERT SPACE PDES

Lemma E.1. *(Infinite Dimensional Feynman-Kac): Define* $\psi : [t_0, T] \times H \to \mathbb{R}$ *as the conditional expectation:*

$$\psi(t,X) := \mathbb{E}_{\mathscr{L}}\left[\exp\Big(-\rho J\big(\mathbb{X}_{t,X}^{T}\big)\Big)\Big|\mathscr{F}_t\right] + \mathbb{E}_{\mathscr{L}}\left[\int_t^T g(X,t)\exp\Big(-\rho\Phi\big(\mathbb{X}_{t,X}^{s}\big)\Big)\mathrm{d}s\Big|\mathscr{F}_t\right], \tag{E.1}$$

evaluated on stochastic trajectories $\mathbb{X}_{t,X}^{T}$ *generated by the infinite dimensional stochastic systems in eqs.* (4.1) *and* (4.2) *of the main text and* $\rho \in \mathbb{R}_+$*. The trajectory dependent terms* $\Phi\big(\mathbb{X}_{t,X}^{T}\big) : L^p \to \mathbb{R}_+$ *and* $J\big(\mathbb{X}_{t,X}^{T}\big) : L^p \to \mathbb{R}_+$ *are defined as follows:*

$$\begin{aligned}\Phi\big(\mathbb{X}_{t,X}^{s}\big) &= \int_t^s \ell\big(\tau, X(\tau)\big)\mathrm{d}\tau,\\ J\big(\mathbb{X}_{t,X}^{T}\big) &= \phi(T,X) + \Phi\big(\mathbb{X}_{t,X}^{T}\big).\end{aligned} \tag{E.2}$$

Also, let $\psi(t,X) \in C_b^{1,2}([0,T]\times H)$*. Then the function* $\psi(t,X)$ *satisfies the following equation:*

$$\begin{aligned}-\partial_t\psi\big(t,X(t)\big) = -\rho\ell\big(t,X(t)\big)\psi\big(t,X(t)\big) + \big\langle\psi_X, \mathscr{A}X(t) + F\big(X(t)\big)\big\rangle\\ + \frac{1}{2}\mathrm{Tr}\Big[\psi_{XX}(BQ^{\frac{1}{2}})(BQ^{\frac{1}{2}})^*\Big] + g\big(t,X(t)\big).\end{aligned} \tag{E.3}$$

Proof. The proof starts with the expectation in eq. (E.1) which is an expectation conditioned on the filtration $\mathscr{F}_t$. To keep the notation short we will drop the dependencies on t and $X(t)$, and will write $\phi_T = \phi\big(T, X(T)\big)$, $\ell_t = \ell\big(t, X(t)\big)$, and $g_t = g\big(t, X(t)\big)$. We split the integrals

inside the expectations to write:

$$\begin{aligned}
\psi(t,X) &= \mathbb{E}_{\mathscr{L}}\left[\exp\left(-\rho\phi_T - \rho\int_t^T \ell_\tau \mathrm{d}\tau\right)\middle|\mathscr{F}_t\right] \\
&\quad + \mathbb{E}_{\mathscr{L}}\left[\int_t^T g_s \exp\left(-\rho\int_t^s \ell_\tau \mathrm{d}\tau\right)\mathrm{d}s\middle|\mathscr{F}_t\right] \\
&= \mathbb{E}_{\mathscr{L}}\left[\exp\left(-\rho\phi_T - \rho\int_{t+\delta t}^T \ell_\tau \mathrm{d}\tau\right)\exp(-\int_t^{t+\delta t} \ell_\tau \mathrm{d}\tau)\middle|\mathscr{F}_t\right] \\
&\quad + \mathbb{E}_{\mathscr{L}}\left[\int_t^{t+\delta t} g_s \exp\left(-\rho\int_t^s \ell_\tau \mathrm{d}\tau\right)\mathrm{d}s\middle|\mathscr{F}_t\right] \\
&\quad + \mathbb{E}_{\mathscr{L}}\left[\int_{t+\delta t}^T g_s \exp\left(-\rho\int_t^s \ell_\tau \mathrm{d}\tau\right)\mathrm{d}s\middle|\mathscr{F}_t\right]
\end{aligned}$$

By using the law of iterated expectations between the two sub-sigma algebras $\mathscr{F}_t \subseteq \mathscr{F}_{t+\delta t}$ we have that:

$$\begin{aligned}
\psi(t,X) = \mathbb{E}_{\mathscr{L}}\Bigg[&\mathbb{E}_{\mathscr{L}}\left[\exp\left(-\rho\phi_T - \rho\int_{t+\delta t}^T \ell_\tau \mathrm{d}\tau\right)\exp\left(-\int_t^{t+\delta t} \ell_\tau \mathrm{d}\tau\right)\middle|\mathscr{F}_{t+\delta t}\right]\middle|\mathscr{F}_t\Bigg] \\
&+ \mathbb{E}_{\mathscr{L}}\left[\int_t^{t+\mathrm{d}t} g_s \exp\left(-\rho\int_t^s \ell_\tau \mathrm{d}\tau\right)\mathrm{d}s\middle|\mathscr{F}_t\right] \\
&+ \mathbb{E}_{\mathscr{L}}\left[\mathbb{E}_{\mathscr{L}}\left[\int_{t+\mathrm{d}t}^T g_s \exp\left(-\rho\int_t^{t+\mathrm{d}t} \ell_\tau \mathrm{d}\tau\right)\exp\left(-\rho\int_{t+\mathrm{d}t}^s \ell_\tau \mathrm{d}\tau\right)\mathrm{d}s\middle|\mathscr{F}_{t+\delta t}\right]\middle|\mathscr{F}_t\right].
\end{aligned}$$

Next we use the fact that the conditioning on the filtration $\mathscr{F}_{t+\delta t}$ results in the following equality:

$$\begin{aligned}
&\mathbb{E}_{\mathscr{L}}\left[\mathbb{E}_{\mathscr{L}}\left[\exp\left(-\rho\phi_T - \rho\int_{t+\delta t}^T \ell_\tau \mathrm{d}\tau\right)\exp\left(-\int_t^{t+\delta t} \ell_\tau \mathrm{d}\tau\right)\middle|\mathscr{F}_{t+\delta t}\right]\middle|\mathscr{F}_t\right] \\
&= \mathbb{E}_{\mathscr{L}}\left[\exp\left(-\int_t^{t+\delta t} \ell_\tau \mathrm{d}\tau\right)\mathbb{E}_{\mathscr{L}}\left[\exp\left(-\rho\phi_T - \rho\int_{t+\delta t}^T \ell_\tau \mathrm{d}\tau\right)\middle|\mathscr{F}_{t+\delta t}\right]\middle|\mathscr{F}_t\right]
\end{aligned}$$

By further using this property of independence we have:

$$\begin{aligned}
\psi(t,X) &= \mathbb{E}_{\mathscr{L}}\left[\exp\left(-\rho\int_t^{t+\delta t}\ell_\tau \mathrm{d}\tau\right)\mathbb{E}_{\mathscr{L}}\left[\exp\left(-\rho\phi_T-\rho\int_{t+\delta t}^{T}\ell_\tau \mathrm{d}\tau\right)\middle|\mathscr{F}_{t+\delta t}\right]\middle|\mathscr{F}_t\right] \\
&\quad+\mathbb{E}_{\mathscr{L}}\left[\int_t^{t+\mathrm{d}t} g_s\exp\left(-\rho\int_t^s\ell_\tau \mathrm{d}\tau\right)\mathrm{d}s\middle|\mathscr{F}_t\right] \\
&\quad+\mathbb{E}_{\mathscr{L}}\left[\exp\left(-\rho\int_t^{t+\delta t}\ell_\tau \mathrm{d}\tau\right)\right]\mathbb{E}_{\mathscr{L}}\left[\int_{t+\delta t}^{T} g_s\exp\left(-\rho\int_{t+\delta t}^s\ell_\tau \mathrm{d}\tau\right)\mathrm{d}s\middle|\mathscr{F}_{t+\delta t}\right]\middle|\mathscr{F}_t\right] \\
&= \mathbb{E}_{\mathscr{L}}\left[\exp\left(-\rho\int_t^{t+\delta t}\ell_\tau \mathrm{d}\tau\right)\psi(t+\delta t,X(t+\delta t))\middle|\mathscr{F}_t\right] \\
&\quad+\mathbb{E}_{\mathscr{L}}\left[\int_t^{t+\delta t} g_s\exp\left(-\rho\int_t^s\ell_\tau \mathrm{d}\tau\right)\mathrm{d}s\middle|\mathscr{F}_t\right]
\end{aligned}$$

The last expression provides the backward propagation of the $\psi\big(t,X(t)\big)$ by employing a expectation over $\psi\big(t+\delta t,X(t+\delta t)\big)$. To get the backward deterministic Kolmogorov equations for the infinite dimensional case we subtract the term $\mathbb{E}\left[\psi\big(t+\delta t,X(t+\delta t)\big)\middle|\mathscr{F}_t\right]$ from both sides:

$$\begin{aligned}
&-\mathbb{E}_{\mathscr{L}}\left[\psi\big(t+\delta t,X(t+\delta t)\big)-\psi\big(t,X(t)\big)\middle|\mathscr{F}_t\right] \\
&\quad=\mathbb{E}_{\mathscr{L}}\left[\left\{\exp\left(-\rho\int_t^{t+\delta t}\ell_\tau \mathrm{d}\tau\right)-1\right\}\psi\big(t+\delta t,X(t+\delta t)\big)\middle|\mathscr{F}_t\right] \\
&\qquad+\mathbb{E}_{\mathscr{L}}\left[\int_t^{t+\delta t} g_s\exp\left(-\rho\int_t^s\ell_\tau \mathrm{d}\tau\right)\mathrm{d}s\middle|\mathscr{F}_t\right].
\end{aligned}$$

Next we take the limit as $\delta t\to 0$ we have:

$$\begin{aligned}
&-\lim_{\delta t\to 0}\mathbb{E}_{\mathscr{L}}\left[\psi\big(t+\delta t,X(t+\delta t)\big)-\psi\big(t,X(t)\big)\middle|\mathscr{F}_t\right] \\
&\quad=\lim_{\delta t\to 0}\mathbb{E}_{\mathscr{L}}\left[\left(\exp(-\rho\int_t^{t+\delta t}\ell_\tau \mathrm{d}\tau)-1\right)\psi\big(t+\delta t,X(t+\delta t)\big)\middle|\mathscr{F}_t\right] \\
&\qquad+\lim_{\delta t\to 0}\mathbb{E}_{\mathscr{L}}\left[\int_t^{t+\delta t} g_s\exp\big(-\rho\int_t^s\ell_\tau \mathrm{d}\tau\big)\mathrm{d}s\middle|\mathscr{F}_t\right].
\end{aligned}$$

Thus we have to compute three terms. We employ the Lebegue dominated convergence theorem to pass the limit inside the expectations:

$$-\lim_{\delta t\to 0}\mathbb{E}_{\mathscr{L}}\left[\psi\big(t+\delta t,X(t+\delta t)\big)-\psi\big(t,X(t)\big)\middle|\mathscr{F}_t\right]=\mathbb{E}_{\mathscr{L}}\left[\mathrm{d}\psi\middle|\mathscr{F}_t\right] \tag{E.4}$$

By using the Itô differentiation rule [70, Theorem 4.32] for the case of infinite dimensional stochastic systems we have that:

$$\mathbb{E}_{\mathscr{L}}\left[\mathrm{d}\psi\big(t,X(t)\big)\middle|\mathscr{F}_t\right] = \partial_t\psi\big(t,X(t)\big)\mathrm{d}t + \big\langle \psi_X, \mathscr{A}X(t) + F\big(X(t)\big)\big\rangle \mathrm{d}t + \frac{1}{2}\mathrm{Tr}\Big[\psi_{XX}(BQ^{\frac{1}{2}})(BQ^{\frac{1}{2}})^*\Big]\mathrm{d}t$$

The next term is

$$\lim_{\delta t\to 0}\mathbb{E}_{\mathscr{L}}\left[\left(\exp(-\rho\int_t^{t+\delta t}\ell_\tau \mathrm{d}\tau) - 1\right)\psi\big(t+\delta t, X(t+\delta t)\big)\middle|\mathscr{F}_t\right] = -\mathbb{E}_{\mathscr{L}}\left[\ell_t\psi\big(t,X(t)\big)\middle|\mathscr{F}_t\right] = -\rho\ell\big(t,X(t)\big)\psi\big(t,X(t)\big)\mathrm{d}t$$

The third term is

$$\lim_{\delta t\to 0}\mathbb{E}_{\mathscr{L}}\left[\int_t^{t+\delta t} g_s \exp\left(-\rho\int_t^s \ell_\tau \mathrm{d}\tau\right)\mathrm{d}s\middle|\mathscr{F}_t\right] = \mathbb{E}_{\mathscr{L}}\left[g\big(t,X(t)\big)\delta t\middle|\mathscr{F}_t\right] = g\big(t,X(t)\big)\mathrm{d}t$$

Combining the three terms above, we have shown that $\psi\big(t,X(t)\big)$ satisfies the backward Kolmogorov equation for the case of the infinite dimensional stochastic system in eq. (4.2) of the main text. □

APPENDIX F

CONNECTIONS TO STOCHASTIC DYNAMIC PROGRAMMING

In this section we show the connections between stochastic dynamic programming and the free energy. Before proceeding, let $C_b^{k,n}([0,T]\times H)$ denote the space of all functions $\xi : [0,T]\times H \to \mathbb{R}^1$ that are k times continuously *Fréchet* differentiable with respect to time t and n times *Gâteaux* differentiable with respect to X. In addition, all their partial derivatives are continuous and bounded in $[0,T]\times H$. Furthermore, trajectories starting at $X \in E$ over the time horizon $[t,T]$ are denoted $\mathbb{X}_{t,X}^T \equiv \mathbb{X}(T,t,\omega;X)$. Using this notation, we have that $\mathbb{X}(t,t,\omega;X) = X$. Finally, for real separable Hilbert space E, by the notation $x\otimes y$ we mean a linear bounded operator on E such that:

$$(x\otimes y)z = x\langle y,z\rangle, \quad \forall x,y,z \in E.$$

First, we perform the exponential transformation on the function $\psi\big(t,X(t)\big) \in C_b^{1,2}([0,T]\times H)$ and show that the transformed function $V\big(t,X(t)\big) \in C_b^{1,2}([0,T]\times H)$ satisfies the HJB equation for the case of infinite dimensional systems [16]. This result is derived with general Q-Wiener noise with covariance operator Q, however it holds also for cylindrical Wiener noise ($Q = I$). This will require applying the Feynman-Kac lemma and deriving the backward Chapman Kolmogorov equation for the case of infinite-dimensional stochastic systems. The backward Kolmogorov equations will result in the HJB equation after a logarithmic transformation is applied. We start from the free energy and relative entropy inequality in eq. (B.2) and define the function $\psi\big(t,X(t)\big) : [0,T]\times H \to \mathbb{R}$ as follows:

$$\psi\big(t,X(t)\big) := \mathbb{E}_{\mathscr{L}}\left[\exp\big(-\rho J\big(\mathbb{X}_{t,X}^T\big)\big)\middle| X\right],$$

which is simply the free energy as defined in definition B.1. By using the Feynman-Kac

lemma we have that the function $\psi(t,X)$ satisfies the backward Chapman Kolmogorov equation specified as follows:

$$\begin{aligned} -\partial_t \psi\big(t,X(t)\big) = -\rho\ell\big(t,X(t)\big)\psi\big(t,X(t)\big) + \big\langle \psi_X, \mathscr{A}X(t) + F\big(X(t)\big)\big\rangle \\ + \frac{1}{2}\mathrm{Tr}\Big[\psi_{XX}(GQ^{\frac{1}{2}})(GQ^{\frac{1}{2}})^*\Big]. \end{aligned} \tag{F.1}$$

where $\partial_t \psi\big(t,X(t)\big)$ denotes the Fréchet derivative of $\psi\big(t,X(t)\big)$ with respect to t, and ψ_X and ψ_{XX} denote the first and second Gâteaux derivatives of $\psi\big(t,X(t)\big)$ with respect to $X(t)$. Starting with the exponential transformation we have:

$$V\big(t,X(t)\big) = -\frac{1}{\rho}\log_e \psi\big(t,X(t)\big) \implies \psi\big(t,X(t)\big) = e^{-\rho V(t,X(t))}.$$

Next we compute the functional derivatives V_X and V_{XX} as functions of the functional derivatives ψ_X and ψ_{XX}. This results in:

$$\begin{aligned} \rho\partial_t V\big(t,X(t)\big)e^{-\rho V} = -\rho\ell\big(t,X(t)\big)e^{-\rho V} - \rho\big\langle V_X e^{-\rho V}, \mathscr{A}X(t) + F\big(X(t)\big)\big\rangle \\ + \frac{\rho}{2}\mathrm{Tr}\Big[(V_X \otimes V_X)(GQ^{\frac{1}{2}})(GQ^{\frac{1}{2}})^* e^{-\rho V}\Big] \\ - \frac{1}{2}\mathrm{Tr}\Big[(V_{XX}(GQ^{\frac{1}{2}})(GQ^{\frac{1}{2}})^* e^{-\rho V}\Big]. \end{aligned}$$

The last equations simplifies to:

$$\begin{aligned} -\partial_t V\big(t,X(t)\big) = \ell\big(t,X(t)\big) + \big\langle V_X, \mathscr{A}X(t) + F\big(X(t)\big)\big\rangle \\ - \frac{1}{2\rho}\mathrm{Tr}\Big[(V_X \otimes V_X)(GQ^{\frac{1}{2}})(GQ^{\frac{1}{2}})^*\Big] + \frac{1}{2\rho}\mathrm{Tr}\Big[V_{XX}(GQ^{\frac{1}{2}})(GQ^{\frac{1}{2}})^*\Big] \end{aligned} \tag{F.2}$$

From the definition of the trace operator $\mathrm{Tr}[A] := \sum_{j=1}^{\infty}\langle Ae_j, e_j\rangle$ for orthonormal basis $\{e_j\}$ over the domain of A, we have the following expression:

$$\frac{1}{2}\mathrm{Tr}\Big[(V_X \otimes V_X)(GQ^{\frac{1}{2\rho}})(GQ^{\frac{1}{2}})^*\Big] = \frac{1}{2\rho}\sum_{j=1}^{\infty}\big\langle (V_X \otimes V_X)(GQ^{\frac{1}{2}})(GQ^{\frac{1}{2}})^* e_j, e_j\big\rangle$$

Since $(x\otimes y)z = x\langle y,z\rangle$ we have that:

$$\begin{aligned}
\frac{1}{2\rho}\sum_{j=1}^{\infty}\big\langle (V_X\otimes V_X)(GQ^{\frac{1}{2}})(GQ^{\frac{1}{2}})^* e_j, e_j\big\rangle &= \frac{1}{2\rho}\sum_{j=1}^{\infty}\Big\langle V_X\big\langle V_X, (GQ^{\frac{1}{2}})(GQ^{\frac{1}{2}})^* e_j\big\rangle, e_j\Big\rangle \\
&= \frac{1}{2\rho}\sum_{j=1}^{\infty}\big\langle V_X, (GQ^{\frac{1}{2}})(GQ^{\frac{1}{2}})^* e_j\big\rangle\big\langle V_X, e_j\big\rangle \\
&= \frac{1}{2\rho}\sum_{j=1}^{\infty}\big\langle (GQ^{\frac{1}{2}})(GQ^{\frac{1}{2}})^* V_X, e_j\big\rangle\big\langle V_X, e_j\big\rangle \\
&\underset{\text{Parseval}}{=} \frac{1}{2}\big\langle V_X, (GQ^{\frac{1}{2}})(GQ^{\frac{1}{2}})^* V_X\big\rangle \\
&= \frac{1}{2\rho}\big|\big|(GQ\frac{1}{2})^* V_X\big|\big|^2_{U_0}
\end{aligned}$$

Substituting back to eq. (F.2) we have the HJB equation for the infinite dimensional case:

$$\begin{aligned}
-V_t\big(t,X(t)\big) = \ell\big(t,X(t)\big) + \big\langle V_X, \mathscr{A}X(t) + F\big(X(t)\big)\big\rangle + \frac{1}{2\rho}\mathrm{Tr}\Big[V_{XX}(GQ^{\frac{1}{2}})(GQ^{\frac{1}{2}})^*\Big] \\
- \frac{1}{2\rho}\big|\big|(GQ^{\frac{1}{2\rho}})^* V_X\big|\big|^2_{U_0}
\end{aligned}$$

In the same vein, one can also show that the relative entropy between the probability measures induced by the uncontrolled and controlled infinite dimensional systems in eqs. (4.1) and (4.2) of the main text, respectively, results in an infinite dimensional quadratic control cost. This requires the use of the Radon-Nikodym derivative from our generalization of Girsanov's theorem for the case of infinite dimensional stochastic systems in eqs. (4.1) and (4.2) of the main text.

APPENDIX G

SPDES UNDER BOUNDARY CONTROL AND NOISE

Let us consider the following problem with Neumann boundary conditions:

$$\begin{cases} \Delta_{\mathsf{x}} y(\mathsf{x}) = \lambda y(\mathsf{x}), \quad \mathsf{x} \in \mathscr{O} \\ \frac{\partial}{\partial n} y(\mathsf{x}) = \gamma(\mathsf{x}), \quad \mathsf{x} \in \partial\mathscr{O} \end{cases} \tag{G.1}$$

where Δ_{x} corresponds to the Laplacian, $\lambda \geq 0$ is a real number, $\mathscr{O}$ is a bounded domain in $\mathbb{R}^d$ with regular boundary $\partial\mathscr{O}$ and $\frac{\partial}{\partial n}$ denotes the normal derivative, with n being the outward unit normal vector. As shown in [16] and references therein, there exists a continuous operator $\mathsf{D}_N : H^s(\partial\mathscr{O}) \to H^{s+3/2}(\mathscr{O})$ such that $\mathsf{D}_N\gamma$ is the solution to eq. (G.1). Given this operator, stochastic parabolic equations with Neumann boundary conditions of the following type:

$$\begin{aligned} &\frac{\partial h(t,\mathsf{x})}{\partial t} = \Delta_{\mathsf{x}} h(t,\mathsf{x}) + f_1(t,h) + c_1(t,h)\frac{\partial w(t,\mathsf{x})}{\partial t}, \quad \mathsf{x} \in \mathscr{O} \\ &\frac{\partial h(t,\mathsf{x})}{\partial n} = f_2(t,h) + c_2(t,h)\frac{\partial v(t,\mathsf{x})}{\partial t}, \quad \mathsf{x} \in \partial\mathscr{O}, \\ &h(0,\mathsf{x}) = h_0(\mathsf{x}). \end{aligned} \tag{G.2}$$

can be written in the mild abstract form:

$$\begin{cases} X(t) = e^{t\mathscr{A}_N} X_0 + \int_0^t e^{(t-s)\mathscr{A}_N} F_1(s,X)\mathrm{d}s + \int_0^t e^{(t-s)\mathscr{A}_N} C_1(s,X)\mathrm{d}W(s) \\ \qquad + \int_0^t (\lambda I - \mathscr{A}_N)^{1/4+\varepsilon} e^{(t-s)\mathscr{A}_N} G_N F_2(s,X)\mathrm{d}s \\ \qquad + \int_0^t (\lambda I - \mathscr{A}_N)^{1/4+\varepsilon} e^{(t-s)\mathscr{A}_N} G_N C_2(s,X)\mathrm{d}V(s), \end{cases} \tag{G.3}$$

where $G_N := (\lambda I - \mathscr{A}_N)^{3/4-\varepsilon}\mathsf{D}_N$, and the remaining terms are defined with respect to the space-time formulation of eq. (G.3). A similar expression can be obtained for Dirichlet

conditions as well, however the solution has to be investigated under weak norms, or in weighted L^2 spaces. More details can be found in [16, Appendix C] and references therein.

APPENDIX H

AN EQUIVALENCE OF THE VARIATIONAL OPTIMIZATION APPROACH FOR SPDES WITH Q-WIENER NOISE

In this section we briefly discuss how one obtains an equivalent variational optimization as in Section III of the main text, for control of SPDEs with Q-Wiener noise. Consider the uncontrolled and controlled version of an H-valued process be given, respectively, by:

$$\mathrm{d}X = \big(\mathscr{A}X + F(t,X)\big)\mathrm{d}t + \frac{1}{\sqrt{\rho}}\sqrt{Q}\mathrm{d}W(t), \tag{H.1}$$

$$\mathrm{d}\tilde{X} = \big(\mathscr{A}\tilde{X} + F(t,\tilde{X})\big)\mathrm{d}t + \sqrt{Q}\big(\mathscr{U}(t,\tilde{X})\mathrm{d}t + \frac{1}{\sqrt{\rho}}\mathrm{d}W(t)\big), \tag{H.2}$$

with initial condition $X(0) = \tilde{X}(0) = \xi$. Here, Q is a trace-class operator, and $W \in U$ is a cylindrical Wiener process. The assumption that Q is of trace class is expressed as:

$$\mathrm{Tr}\big[Q\big] = \sum_{n=1}^{\infty} \big\langle Qe_n, e_n \big\rangle < \infty.$$

As opposed to the discussion following eq. (2.13) of the main text, in this case we do not require any contractive assumption on the operator $\mathscr{A}$ due to the nuclear property of the operator Q. The stochastic integral $\int_0^t e^{(t-s)\mathscr{A}}\sqrt{Q}\mathrm{d}W(s)$ is well defined in this case [70, Chapter 4.2]. Define the process:

$$\begin{aligned} W_Q(t) &:= \sqrt{Q}W(t) = \sum_{n=1}^{\infty} \sqrt{Q}e_n\beta_n(t) \\ &= \sum_{n=1}^{\infty} \sqrt{\lambda_n}e_n\beta_n(t) \end{aligned}$$

where the basis $\{e_n\}$ satisfies the eigenvalue-eigenvector relationship $Qe_n = \lambda e_n$. The

process $W_Q(t)$ satisfies the properties in Definition A.4, and is therefore a Q-Wiener process.

The above case is an SPDE driven by Q-Wiener noise, which is quite different from the cylindrical Wiener process described in the rest of this work. In order to state the Girsanov's theorem in this case, we first define the Hilbert space $U_0 := \sqrt{Q}(U) \subset U$ with inner product $\langle u, v\rangle_{U_0} := \left\langle Q^{-1/2}u, Q^{-1/2}v\right\rangle_U$, $\forall u, v \in U_0$.

Theorem H.1 (Girsanov)**.** *Let Ω be a sample space with a σ-algebra $\mathscr{F}$. Consider the following H-valued stochastic processes:*

$$\mathrm{d}X = \big(\mathscr{A}X + F(t,X)\big)\mathrm{d}t + \frac{1}{\sqrt{\rho}}\mathrm{d}W_Q(t), \tag{H.3}$$

$$\mathrm{d}\tilde{X} = \big(\mathscr{A}\tilde{X} + F(t,\tilde{X})\big)\mathrm{d}t + \sqrt{Q}\mathscr{U}(t,\tilde{X})\mathrm{d}t + \frac{1}{\sqrt{\rho}}\mathrm{d}W_Q(t)\big), \tag{H.4}$$

where $X(0) = \tilde{X}(0) = x$ and $W_Q \in U$ is a Q-Wiener process with respect to measure $\mathbb{P}$. Moreover, for each $\Gamma \in C([0,T];H)$, let the law of X be defined as $\mathscr{L}(\Gamma) := \mathbb{P}(\omega \in \Omega | X(\cdot,\omega) \in \Gamma)$. Similarly, the law of $\tilde{X}$ is defined as $\tilde{\mathscr{L}}(\Gamma) := \mathbb{P}(\omega \in \Omega | \tilde{X}(\cdot,\omega) \in \Gamma)$. Then

$$\tilde{\mathscr{L}}(\Gamma) = \mathbb{E}_{\mathbb{P}}\big[\exp\big(\int_0^T \left\langle \psi(s), dW_Q(s)\right\rangle_{U_0} - \tfrac{1}{2}\int_0^T ||\psi(s)||^2_{U_0} ds\big) | X(\cdot) \in \Gamma\big], \tag{H.5}$$

where we have defined $\psi(t) := \sqrt{\rho}\mathscr{U}\left(t, \tilde{X}(t)\right) \in U_0$ and assumed

$$\mathbb{E}_{\mathbb{P}}\left[e^{\frac{1}{2}\int_0^T ||\psi(t)||^2 \mathrm{d}t}\right] < +\infty. \tag{H.6}$$

Proof. The proof is identical to the proof of theorem C.1. □

Note that $\psi(t)$ in this case is identical to $\psi(t)$ in Theorem Theorem C.1. As a result, despite having Q-Wiener noise, we have the same variational optimization for this case as in Section III of the main text.

APPENDIX I

A COMPARISON TO VARIATIONAL OPTIMIZATION IN FINITE DIMENSIONS

In what follows we show how degeneracies arise for a similar derivation in finite dimensions. The stochastic dynamics are given by:

$$\mathrm{d}X = \big(\mathscr{A}X + F(t,X)\big)\mathrm{d}t + G(t,X)\big(\mathscr{U}(t,X)\mathrm{d}t + \frac{1}{\sqrt{\rho}}\mathrm{d}W(t)\big), \tag{I.1}$$

where W(t) is a cylindrical Wiener process. Now, let the Hilbert space state vector $X(t) \in H$ be approximated by a finite dimensional state vector $X(t) \approx \hat{X}(t) \in \mathbb{R}^n$ with arbitrary accuracy, where n is the number of grid points. In order to rewrite a finite dimensional form of eq. (I.1), the cylindrical Wiener noise term $W(t)$ must be captured by a finite dimensional approximation. The expansion of $W(t)$ in eq. (A.1) is restated here and truncated at m terms:

$$W(t) = \sum_{j=1}^{\infty} \sqrt{\lambda_j}\beta_j(t)e_j = \sum_{j=1}^{\infty} \beta_j(t)e_j \approx \sum_{j=1}^{m} \beta_j(t)e_j \tag{I.2}$$

where $\lambda_j = 1, \forall j \in \mathbb{N}$ in the case of cylindrical Wiener noise, and $\beta_j(t)$ is a standard Wiener process on $\mathbb{R}$. The stochastic dynamics in eq. (I.1) become a finite set of SDEs:

$$\mathrm{d}\hat{X} = \big(\mathscr{A}\hat{X} + \mathscr{F}(t,\hat{X})\big)\mathrm{d}t + \mathscr{G}(t,\hat{X})\big(\mathscr{M}\mathbf{u}(t;\theta)\mathrm{d}t + \frac{1}{\sqrt{\rho}}\mathscr{R}\mathrm{d}\beta(t)\big) \tag{I.3}$$

The terms $\mathscr{A}$, $\mathscr{F}$, and $\mathscr{G}$ are matrices associated with the Hilbert space operators $\mathscr{A}$, F, and G respectively. The matrix $\mathscr{M}$ has dimensionality $\mathscr{M} \in \mathbb{R}^{n\times k}$, where k is the number of actuators placed in the field. The vector $\mathrm{d}\beta \in \mathbb{R}^m$ collects the Wiener noise terms in the expansion eq. (I.2), and the matrix $\mathscr{R}$ collects finite dimensional basis vectors from eq. (I.2). As noted in section 4.4, the dimensionality of the $\mathscr{R}$ is $\mathscr{R} \in \mathbb{R}^{n\times m}$. The degeneracy arises

when $n > m$ for the case of the cylindrical noise. For the case of Q-Wiener noise, degeneracy may arises even when $n \leq m$ and $\text{Rank}(\mathscr{R}) < n$. In both cases, the issue of degeneracy prohibits the use of Girsanov theorem for the importance sampling steps due to the lack of invertibility of $\mathscr{R}$. With respect to the approach relying on Gaussian densities, the derivation would require the following time discretization of the reduced order model in eq. (I.3):

$$\hat{X}(t+\Delta t) = \hat{X}(t) + \int_t^{t+\Delta t} \Big(\mathscr{A}\hat{X} + \mathscr{F}(t,\hat{X})\Big)\mathrm{d}t + \int_t^{t+\Delta t} \mathscr{G}(t,\hat{X})\Big(\mathscr{M}\mathbf{u}(t;\theta)\mathrm{d}t + \frac{1}{\sqrt{\rho}}\mathscr{R}\mathrm{d}\beta(t)\Big) \tag{I.4}$$

$$\approx \hat{X}(t) + \Big(\mathscr{A}\hat{X} + \mathscr{F}(t,\hat{X})\Big)\Delta t + \mathscr{G}(t,\hat{X})\Big(\mathscr{M}\mathbf{u}(t;\theta)\Delta t + \frac{1}{\sqrt{\rho}}\mathscr{R}\mathrm{d}\beta(t)\Big) \tag{I.5}$$

$$\tag{I.6}$$

Without loss of generality we simplify the expression above by assuming the $\mathscr{G}(t,\hat{X}) = I_{n\times n}$. The transition probability will take the following form:

$$\begin{aligned}&\mathrm{p}\big(\hat{X}(t+\Delta t)|\hat{X}(t)\big)\\ &= \frac{1}{(\sqrt{2\pi})^n(\det\Sigma_{\hat{X}})^{\frac{1}{2}}}\exp\Bigg(-\frac{1}{2}\Big(\hat{X}(t+\Delta t)-\mu_{\hat{X}}(t+\Delta t)\Big)^{\top}\Sigma_{\hat{X}}^{-1}\Big(\hat{X}(t+\Delta t)-\mu_{\hat{X}}(t+\Delta t)\Big)\Bigg)\end{aligned} \tag{I.7}$$

where the term $\mu_{\hat{X}}(t+\Delta t)$ is the mean and $\Sigma_{\hat{X}}$ is the variance defined as follows:

$$\mu_{\hat{X}}(t+\Delta t) = \hat{X}(t) + \big(\mathscr{A}\hat{X} + \mathscr{F}(t,\hat{X})\big)\Delta t + \mathscr{M}\mathbf{u}(t;\theta)\Delta t \tag{I.8}$$

$$\Sigma_{\hat{X}} = \frac{1}{\rho}\mathscr{R}\mathscr{R}^{\mathrm{T}}\Delta t \tag{I.9}$$

The existence of the transition probability densities requires invertibility of $\mathscr{R}\mathscr{R}^{\mathrm{T}}$ which is not possible when $n < m$ or when $\text{Rank}(\mathscr{R}) < n$ for $n \geq m$.

APPENDIX J

BRIEF DESCRIPTION OF OPEN LOOP AND MPC EXPERIMENTS

The following is additional information about the experiments referenced in Section V. Section section J.1 describes boundary and distributed control experiments, while Sections section J.2 and section J.3 describe experiments for distributed control only.

J.1 Heat SPDE

The 2D stochastic Heat PDE with homogeneous Dirichlet boundary conditions given by:

$$\begin{aligned} h_t(t,x,y) &= \varepsilon h_{xx}(t,x,y) + \varepsilon h_{yy}(t,x,y) + \sigma dW(t), \\ h(t,0,y) &= h(t,a,y) = h(t,x,0) = h(t,x,a) = 0, \\ h(0,x,y) &\sim \mathcal{N}(h_0; 0, \sigma_0), \end{aligned} \tag{J.1}$$

where the parameter ε is the so called thermal diffusivity, which governs how quickly the initial temperature profile diffuses across the spatial domain. Equation (J.1) considers the scenario of controlling a metallic plate to a desired temperature profile using 5 actuators distributed across the plate. The edges of the plate are always held at constant temperature of 0 degrees Celsius. The parameter a is the length of the sides of the square plate, for which we use $a = 0.5$ meters.

The actuator dynamics are modeled by Gaussian-like exponential functions with the means co-located with the actuator locations at: $\mu = \left[\mu_1, \mu_2, \mu_3, \mu_4, \mu_5\right] = \big[(0.2a, 0.5a)$, $(0.5a, 0.2a), (0.5a, 0.5a), (0.5a, 0.8a), (0.8a, 0.5a)\big]$ and the variance of the effect of each actuator on nearby field states given by $\sigma_l^2 = (0.1a)^2$, $\forall l = 1, \ldots, 5$. For every $j = 1, \ldots, J$,

and $l = 1, \ldots, N$, the resulting $m_l(\mathbf{x})$ has the form:

$$m_{l,j}\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \exp\left\{-\frac{1}{2}\left(\begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} \mu_{l,x} \\ \mu_{l,y} \end{bmatrix}\right)^{\top} \begin{bmatrix} \sigma_l^2 & 0 \\ 0 & \sigma_l^2 \end{bmatrix} \left(\begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} \mu_{l,x} \\ \mu_{l,y} \end{bmatrix}\right)\right\}$$

The spatial domain is discretized by dividing the x and y domains into 64 points each creating a grid of 64×64 spatial locations on the plate surface. For our experiments, we use a semi-implicit forward Euler discretization scheme for time and central difference for the 2^{nd} order spatial derivatives h_{xx} and h_{yy}. We used the following parameter values, time discretization $\Delta t = 0.01s$, MPC time horizon $T = 0.05s$, total simulation time $T_{\text{sim}} = 1.0s$, thermal diffusivity $\varepsilon = 1.0$ and initialization standard deviation $\sigma_0 = 0.5$. The cost function considered for the experiments was defined as follows:

$$J := \sum_t \sum_x \sum_y \kappa\big(h_{\text{actual}}(t,x,y) - h_{\text{desired}}(t,x,y)\big)^2 \cdot \mathbb{1}_S(x,y)$$

where $S := \cup_{i=1}^{5} S_i$ and the indicator function $\mathbb{1}_S(x,y)$ is defined as follows:

$$\mathbb{1}_S(x,y) := \begin{cases} 1, & \text{if} \quad (x,y) \in S \\ 0, & \text{otherwise} \end{cases} \tag{J.2}$$

where

$S_1 = \{(x,y) \mid x \in [0.48a, 0.52a] \text{ and } y \in [0.48a, 0.52a]\}$ is in the central region of the plate

$S_2 = \{(x,y) \mid x \in [0.22a, 0.18a] \text{ and } y \in [0.48a, 0.52a]\}$ is the left-mid region of the plate

$S_3 = \{(x,y) \mid x \in [0.82a, 0.78a] \text{ and } y \in [0.48a, 0.52a]\}$ is the right-mid region of the plate

$S_4 = \{(x,y) \mid x \in [0.48a, 0.52a] \text{ and } y \in [0.18a, 0.22a]\}$ is in the top-central region of the plate

$S_5 = \{(x,y) \mid x \in [0.48a, 0.52a] \text{ and } y \in [0.78a, 0.82a]\}$ is in the bottom-central region of the plate

In addition $h_{\text{desired}}(t,x,y) = 0.5^{\circ}C$ for $(x,y) \in S_1$ and $h_{\text{desired}}(t,x,y) = 1.0^{\circ}C$ for $(x,y) \in \cup_{i=2}^{5} S_i$ and the scaling parameter $\kappa = 100$.

In the boundary control case, we make use of the 1D stochastic heat equation given as follows:

$$h_t(t,x) = \varepsilon h_{xx}(t,x) + \sigma dW(t)$$
$$h(0,x) = h_0(x)$$

For Dirichlet and Neumann boundary conditions we have $h(t,x) = \gamma(x)$, $\forall x \in \partial O$ and $h_x(t,x) = \gamma(x)$, $\forall x \in \partial O$, respectively. Regarding our 1-D boundary control example, we set $\varepsilon = 1$, $\sigma = 0.1$, $h_x(t,0) = u_1(t)$ and $h_x(t,a) = u_2(t)$. In this case, $m_l(x)$ is simply given by the identity function and the corresponding inner products associated with Girsanov's theorem are given by the standard dot product. Finally, the cost function used is the same as above with $S = \{x | 0 < x < a\}$ and

$$h_{\text{desired}}(t,x) = \begin{cases} 1, & \text{for} \quad t \in [0,0.4], \\ 3, & \text{for} \quad t \in [0,0.4] \text{ and } t \in [0.8,1.3]. \end{cases}$$

J.2 Burgers SPDE

The 1D stochastic Burgers PDE with non-homogeneous Dirichlet boundary conditions is as follows:

$$\begin{aligned} h_t(t,x) + hh_x(t,x) &= \varepsilon h_{xx}(t,x) + \sigma dW(t) \\ h(t,0) &= h(t,a) = 1.0 \\ h(0,x) &= 0, \ \forall x \in (0,a) \end{aligned} \tag{J.3}$$

where the parameter ε is the viscosity of the medium. Equation (J.3) considers a simple model of a 1D flow of a fluid in a medium with non-zero flow velocities at the two boundaries.

The goal is to achieve and maintain a desired flow velocity profile at certain points along the spatial domain. As seen in the desired profile in Fig. 3 of the main paper, there are 3 areas along the spatial domain with desired flow velocity such that the flow has to be accelerated, then decelerated, and then accelerated again while trying to overcome the stochastic forces and the dynamics governed by the Burgers PDE. Similar to the experiments for the Heat SPDE, we consider actuators behaving as Gaussian-like exponential functions with the means co-located with the actuator locations at: $\mu = \big[0.2a, 0.3a, 0.5a, 0.7a, 0.8a\big]$ and the spatial effect (variance) of each actuator given by $\sigma_l^2 = (0.1a)^2, \forall\, l = 1, \ldots, 5$. The parameter $a = 2.0\,m$ is the length of the channel along which the fluid is flowing.

This spatial domain was discretized using a grid of 128 points. The numerical scheme used semi-implicit forward Euler discretization for time and central difference approximation for both the 1^{st} and 2^{nd} order derivatives in space. The 1^{st} order derivative terms in the advection term hh_x were evaluated at the current time instant while the 2^{nd} order spatial derivatives in the diffusion term h_{xx} were evaluated at the next time instant, hence the scheme is semi-implicit. Following are values of some other parameters used in our experiments: time discretization $\Delta t = 0.01$, total simulation time = $1.0\,s$, MPC time horizon = $0.1\,s$, and the scaling parameter $\kappa = 100$. The cost function considered for the experiments was defined as follows:

$$J := \sum_t \sum_x \kappa \big(h_{\text{actual}}(t,x) - h_{\text{desired}}(t,x)\big)^2 \cdot \mathbb{1}_S(x)$$

where the function $\mathbb{1}_S(x)$ is defined as in eq. (J.2) with $S = \cup_{i=1}^{3}$, where $S_1 = [0.18a, 0.22a]$, $S_2 = [0.48a, 0.52a]$, and $S_3 = [0.78a, 0.82a]$. In addition $h_{\text{desired}}(t,x) = 2.0\ m/s$ for $x \in S_1 \cup S_3$ which is at the sides, and $h_{\text{desired}}(t,x) = 1.0\ m/s$ for $x \in S_2$ which is in the central region.

J.3 Nagumo SPDE

The stochastic Nagumo equation with Neumann boundary conditions is as follows:

$$
\begin{aligned}
h_t(t,x) &= \varepsilon h_{xx}(t,x) + h(t,x)\big(1-h(t,x)\big)\big(h(t,x)-\alpha\big) + \sigma dW(t) \\
h_x(t,0) &= h_x(t,a) = 0 \\
h(0,x) &= \Big(1+\exp(-\frac{2-x}{\sqrt{2}})\Big)^{-1}
\end{aligned}
$$

The parameter α determines the speed of a wave traveling down the length of the axon and ε the rate of diffusion. By simulating the deterministic Nagumo equation with $a = 5.0$, $\varepsilon = 1.0$ and $\alpha = -0.5$, we observed that after about 5 seconds, the wave completely propagates to the end of the axon. Similar to the experiments for the Heat SPDE, we consider actuators behaving as Gaussian-like exponential functions with actuator centers (mean values) at $\mu = \big[0.2a, 0.3a, 0.4a, 0.5a, 0.6a, 0.7a, 0.8a\big]$ and the spatial effect (variance) of each actuator given by $\sigma_l^2 = (0.1a)^2, \forall\, l = 1, \ldots, 7$. The spatial domain was discretized using a grid of 128 points. The numerical scheme used semi-implicit forward Euler discretization for time and central difference approximation for the 2^{nd} order derivatives in space. Following are values of some other parameters used in our experiments: time discretization $\Delta t = 0.01$, MPC time horizon = $0.1\,s$, total simulation time = $1.5\,s$ for acceleration task and total simulation time = $5.0\,s$ for the suppression task, and the scaling parameter $\kappa = 10000$. The cost function for this experiment was defined as follows:

$$
J = \sum_t \sum_x \kappa\big(h_{\text{actual}}(t,x) - h_{\text{desired}}(t,x)\big)^2 \cdot \mathbb{1}_S(x)
$$

where $h_{\text{desired}}(t,x) = 0.0\,V$ for the suppression task, and $h_{\text{desired}}(t,x) = 1.0\,V$ for the acceleration task, and the function $\mathbb{1}_S(x)$ is defined as in eq. (J.2) with $S = [0.7a, 0.99a]$.

APPENDIX K

DERIVATION OF VARIATIONAL MINIMIZATION AND LOSS FUNCTION

This section explains the steps to arrive at eqs. (5.4), (5.6) and (5.7) from the main paper.

$$\begin{aligned}
\Theta^* &= \underset{\Theta}{\operatorname{argmin}} D_{KL}(\mathscr{L}^* || \tilde{\mathscr{L}}) \\
&= \underset{\Theta}{\operatorname{argmin}} \left[\int \log\left(\frac{\mathrm{d}\mathscr{L}^*}{\mathrm{d}\tilde{\mathscr{L}}}\right) \mathrm{d}\mathscr{L}^* \right] \\
&= \underset{\Theta}{\operatorname{argmin}} \left[\int \log\left(\frac{\mathrm{d}\mathscr{L}^*}{\mathrm{d}\mathscr{L}} \frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}}\right) \mathrm{d}\mathscr{L}^* \right] \\
&= \int \log\left(\frac{\mathrm{d}\mathscr{L}^*}{\mathrm{d}\mathscr{L}}\right) \mathrm{d}\mathscr{L}^* + \underset{\Theta}{\operatorname{argmin}} \left[\int \log\left(\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}}\right) \mathrm{d}\mathscr{L}^* \right] \\
&= \underset{\Theta}{\operatorname{argmin}} \left[\int \log\left(\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}}\right) \frac{\mathrm{d}\mathscr{L}^*}{\mathrm{d}\mathscr{L}} \frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}} \mathrm{d}\tilde{\mathscr{L}} \right] = \underset{\Theta}{\operatorname{argmin}}\ L
\end{aligned} \tag{K.1}$$

Now,

$$L = \mathbb{E}_{\tilde{\mathscr{L}}} \left[\log\left(\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}}\right) \frac{\mathrm{d}\mathscr{L}^*}{\mathrm{d}\mathscr{L}} \frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}} \mathrm{d}\tilde{\mathscr{L}} \right]$$

Substituting eq. (5.5), the log and exponential cancel,

$$\begin{aligned}
L = \mathbb{E}_{\tilde{\mathscr{L}}} \Bigg[\Bigg(& -\sqrt{\rho} \int_0^T \left\langle \Phi(t, X, \mathbf{x}; \Theta^{(k)}), \mathrm{d}W(t) \right\rangle \\
& - \frac{1}{2}\rho \int_0^T \left\langle \Phi(t, X, \mathbf{x}; \Theta^{(k)}), \Phi(t, X, \mathbf{x}; \Theta^{(k)}) \right\rangle \mathrm{d}t \Bigg) \frac{\mathrm{d}\mathscr{L}^*}{\mathrm{d}\mathscr{L}} \frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}} \mathrm{d}\tilde{\mathscr{L}} \Bigg]
\end{aligned}$$

Evaluating, $\frac{\mathrm{d}\mathscr{L}^*}{\mathrm{d}\mathscr{L}}\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}}$ separately, we have,

$$\begin{aligned}\frac{\mathrm{d}\mathscr{L}^*}{\mathrm{d}\mathscr{L}}\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}} &= \frac{\exp(-\rho J)}{\mathbb{E}_{\mathscr{L}}\big[\exp(-\rho J)\big]}\exp\Bigg(-\sqrt{\rho}\int_0^T\Big\langle\Phi(t,X,\mathbf{x};\Theta^{(k)}),\mathrm{d}W(t)\Big\rangle \\ &\qquad -\frac{1}{2}\rho\int_0^T\Big\langle\Phi(t,X,\mathbf{x};\Theta^{(k)}),\Phi(t,X,\mathbf{x};\Theta^{(k)})\Big\rangle\mathrm{d}t\Bigg) \\ &= \frac{\exp(-\rho\tilde{J})}{\mathbb{E}_{\mathscr{L}}\big[\exp(-\rho J)\big]},\end{aligned}$$

where $\tilde{J}$ is defined in eq. (5.7). Similarly, we can use importance sampling for the expectation in the denominator using eq. (5.5) as,

$$\mathbb{E}_{\mathscr{L}}\big[\exp(-\rho J)\big] = \mathbb{E}_{\tilde{\mathscr{L}}}\left[\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}}\exp(-\rho J)\right] = \mathbb{E}_{\tilde{\mathscr{L}}}\big[\exp(-\rho\tilde{J})\big]$$

Thus, we have

$$\frac{\mathrm{d}\mathscr{L}^*}{\mathrm{d}\mathscr{L}}\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}} = \frac{\exp(-\rho\tilde{J})}{\mathbb{E}_{\tilde{\mathscr{L}}}\big[\exp(-\rho\tilde{J})\big]}$$

Putting all of this together, we get the required form of eq. (5.6) as,

$$\begin{aligned}L = \mathbb{E}_{\tilde{\mathscr{L}}}\Bigg[&\frac{\exp(-\rho\tilde{J})}{\mathbb{E}_{\tilde{\mathscr{L}}}\big[\exp(-\rho\tilde{J})\big]}\Bigg(-\sqrt{\rho}\int_0^T\Big\langle\Phi(t,X,\mathbf{x};\Theta^{(k)}),\mathrm{d}W(t)\Big\rangle \\ &-\frac{1}{2}\rho\int_0^T\Big\langle\Phi(t,X,\mathbf{x};\Theta^{(k)}),\Phi(t,X,\mathbf{x};\Theta^{(k)})\Big\rangle\mathrm{d}t\Bigg)\Bigg]\end{aligned}$$

APPENDIX L

ADDITIONAL INFORMATION ON IDVRL SIMULATIONS

Following are some details on each of our simulations which will help in reproducing our results.

L.1 1D Heat SPDE distributed and boundary control

L.1.1 Distributed Control

The heat SPDE in 1D is given by

$$\begin{aligned} dh(t,x) &= \varepsilon h_{xx}(t,x)dt + G(t,h)\big(\mathbf{m}(\mathbf{x})^\top \varphi(h;\Theta)dt + \sigma dW(t)\big) \\ h(0,x) &= h_0(x) \end{aligned} \tag{L.1}$$

where ε is the thermal diffusivity parameter, which was set to 1 for our experiments. The task is to achieve a desired temperature profile at 3 regions along the spatial domain. At the center of these regions are actuators. The three-actuator-based control is achieved by setting $\mathbf{m}(\mathbf{x})^\top = [m_1(\mathbf{x}), m_2(\mathbf{x}), m_3(\mathbf{x})]^\top$ and $G(t,h)$ to an identity operator. The actuator dynamics $m(\mathbf{x})$ are modelled by Gaussian-like exponential functions with the means co-located with the actuator locations at: $\mu = \big[\mu_1, \mu_2, \mu_3\big] = \big[0.2a, 0.5a, 0.8a\big]$ and the variance of the effect of each actuator on nearby field states given by $\sigma_l^2 = (0.1a)^2, \forall l = 1,2,3$. The cost function considered for the experiments is defined as

$$J := \sum_t \sum_x \kappa\big(h_{\text{actual}}(t,x) - h_{\text{desired}}(t,x)\big)^2 \cdot \mathbb{1}_S(x) \tag{L.2}$$

where $S := \cup_{i=1}^{3} S_i$ and the indicator function $\mathbb{1}_S(x)$ is defined as

$$\mathbb{1}_S(x) := \begin{cases} 1, & \text{if } x \in S \\ 0, & \text{otherwise} \end{cases} \tag{L.3}$$

where,

$$\begin{aligned} &S_1 = \{x \in D \mid x \in [0.18a, 0.22a]\} \text{ is the region of the spatial domain on the left,} \\ &S_2 = \{x \in D \mid x \in [0.48a, 0.52a]\} \text{ is the region of the spatial domain in the center,} \\ &S_3 = \{x \in D \mid x \in [0.78a, 0.82a]\} \text{ is the region of the spatial domain on the right.} \end{aligned} \tag{L.4}$$

The non-linear policy $\varphi(h;\Theta)$ was chosen to be a FNN with 2 hidden layers of 64 neurons each and ReLU activations. The network was trained using the ADAM optimizer for 1000 iterations with 200 trajectories sampled from the Heat SPDE model per iteration. Each trajectory was 1.0 seconds long with $\Delta t = 0.01$ seconds.

These parameters were run over 200 trials to obtain the convergence results depicted in fig. 5.4. These plots demonstrate that even though the state cost is not monotonically decreasing, the loss has a monotonic-like decreasing behavior. As described in the main text, this demonstrates that IDVRL may be pushing the state cost out of local minima. Additionally, the variance in the algorithm decreases over iterations.

L.1.2 Boundary Control

In the boundary control case, we make use of the 1D stochastic heat equation

$$\begin{aligned} dh(t,x) &= \varepsilon h_{xx}(t,x)dt + G(t,h)\left(\mathbf{m}(\mathbf{x})^\top \varphi(h;\Theta)dt + \sigma dW(t)\right) \\ h(0,x) &= h_0(x) \end{aligned} \tag{L.5}$$

For Dirichlet and Neumann boundary conditions we have $h(t,x) = \gamma(x)$, $\forall x \in \partial O$

and $h_x(t,x) = \gamma(x)$, $\forall x \in \partial O$, respectively. In our 1-D boundary control example, we set $\varepsilon = 1$, $\rho = 10$, $h_x(t,0) = u_1(t) + \frac{1}{\sqrt{\rho}} dW(t)$ and $h_x(t,a) = u_2(t) + \frac{1}{\sqrt{\rho}} dW(t)$. In the infinite-dimensional Hilbert space formulation, these boundary conditions are incorporated into the $G(t,h)\,\varphi(h;\Theta)$ term. In this case, $\mathbf{m}(\mathbf{x})^\top = [m_1(\mathbf{x}), m_2(\mathbf{x})]^\top$, where each $m(\mathbf{x})$ is simply given by an indicator function and $G(t,h)$ is an identity operator. The cost function is given by eq. (L.2) with $S = D$ and $h_{desired}(t,x) = 3$.

For 1D boundary control, the non-linear policy $\varphi(h;\Theta)$ was chosen to be a FNN with 2 hidden layers of 64 neurons each and ReLU activations. The network was trained using the ADAM optimizer for 1000 iterations with 200 trajectories sampled from the Heat SPDE model per iteration. Each trajectory was 1.5 seconds long with $\Delta t = 0.01$ seconds.

L.2 2D Heat SPDE distributed control

The 2D Heat SPDE with homogeneous Dirichlet boundary conditions given by

$$\begin{aligned} dh(t,x,y) &= \varepsilon h_{xx}(t,x,y)dt + \varepsilon h_{yy}(t,x,y)dt + G(t,h)\big(\mathbf{m}(\mathbf{x})^\top \varphi(h;\Theta)dt + \sigma dW(t)\big), \\ h(t,0,y) &= h(t,a,y) = h(t,x,0) = h(t,x,a) = 0, \\ h(0,x,y) &\sim \mathscr{N}(h_0;0,\sigma_0), \end{aligned} \tag{L.6}$$

where the parameter ε is the so called thermal diffusivity, which governs how quickly the initial temperature profile diffuses across the spatial domain. Equation (L.6) considers the scenario of controlling a metallic plate to a desired temperature profile using 5 actuators distributed across the plate. The edges of the plate are always held at constant temperature of 0 degrees Celsius. The parameter a is the length of the sides of the square plate, for which we use $a = 0.25$ meters.

The 5 actuator-based control is achieved by setting $\mathbf{m}(\mathbf{x})^\top = [m_1(\mathbf{x}), m_2(\mathbf{x}), m_3(\mathbf{x}),$ $m_4(\mathbf{x}), m_5(\mathbf{x})]^\top$ and $G(t,h)$ to an identity operator. The actuator dynamics $m(\mathbf{x})$ are modelled by Gaussian-like exponential functions with the means co-located with the actuator locations at: $\mu = \big[\mu_1, \mu_2, \mu_3, \mu_4, \mu_5\big] = \big[(0.2a, 0.5a), (0.5a, 0.2a), (0.5a, 0.5a), (0.5a, 0.8a),$

$(0.8a, 0.5a)\big]$ and the variance of the effect of each actuator on nearby field states given by $\sigma_l^2 = (0.1a)^2$, $\forall l = 1,\ldots,5$. The spatial domain is discretized by dividing the x and y domains into $J = 32$ points each creating a grid of 32×32 spatial locations on the plate surface. The resulting $m_l(\mathbf{x})$ has the form

$$m_{l,j}\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \exp\left\{-\frac{1}{2}\left(\begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} \mu_{l,x} \\ \mu_{l,y} \end{bmatrix}\right)^{\top} \begin{bmatrix} \sigma_l^2 & 0 \\ 0 & \sigma_l^2 \end{bmatrix} \left(\begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} \mu_{l,x} \\ \mu_{l,y} \end{bmatrix}\right)\right\},$$
$$\forall\, j = 1,\ldots,J,\ \ l = 1,\ldots,5$$

For our simulations, we use a semi-implicit forward Euler discretization scheme for time and central difference for the 2^{nd} order spatial derivatives h_{xx} and h_{yy}. We used time discretization $\Delta t = 0.02s$, simulation time horizon $T = 1.0s$ and thermal diffusivity $\varepsilon = 1.0$. The cost function considered for the experiments is defined as

$$J := \sum_t \sum_x \sum_y \kappa\big(h_{\text{actual}}(t,x,y) - h_{\text{desired}}(t,x,y)\big)^2 \cdot \mathbb{1}_S(x,y)$$

where $S := \cup_{i=1}^{5} S_i$ and the indicator function $\mathbb{1}_S(x,y)$ is defined similar to eq. (L.3) as

$$\mathbb{1}_S(x,y) := \begin{cases} 1, & \text{if } (x,y) \in S \\ 0, & \text{otherwise,} \end{cases}$$

where,

$S_1 = \{(x,y) \in D \mid x \in [0.48a, 0.52a] \text{ and } y \in [0.48a, 0.52a]\}$ is in the central region,

$S_2 = \{(x,y) \in D \mid x \in [0.22a, 0.18a] \text{ and } y \in [0.48a, 0.52a]\}$ is the left-mid region,

$S_3 = \{(x,y) \in D \mid x \in [0.82a, 0.78a] \text{ and } y \in [0.48a, 0.52a]\}$ is the right-mid region,

$S_4 = \{(x,y) \in D \mid x \in [0.48a, 0.52a] \text{ and } y \in [0.18a, 0.22a]\}$ is in the top-central region,

$S_5 = \{(x,y) \in D \mid x \in [0.48a, 0.52a] \text{ and } y \in [0.78a, 0.82a]\}$ is in the bottom-central region.

Table L.1: Description of CNN policy network for 2D Heat SPDE.

Layer name	Kernel size	# Filters (output size)	Stride	Padding type	Activation
Input	-	1	-	-	-
Conv-1	4	5	2	VALID	ReLU
Max-pool-1	2	-	2	-	-
Conv-2	2	16	1	SAME	ReLU
Max-pool-2	2	-	2	-	-
Dense	-	5	-	-	Linear

In addition $h_{\text{desired}}(t,x,y) = 0.5^{\circ}C$ for $(x,y) \in S_1$ and $h_{\text{desired}}(t,x,y) = 1.0^{\circ}C$ for $(x,y) \in \cup_{i=2}^{5} S_i$ and the scaling parameter $\kappa = 10^{-3}$.

Since the domain is 2D, the inputs to the non-linear policy $\varphi(h;\Theta)$ are image-like data after discretization, and therefore the policy was chosen to be a CNN. The description of the network architecture is given in table L.1. The network was trained using the ADAM optimizer for 1000 iterations with 50 trajectories sampled from the 2D Heat SPDE model per iteration. Each trajectory was 1.0 seconds long with $\Delta t = 0.02$.

L.3 1D Burgers SPDE distributed control

The 1D Burgers SPDE with non-homogeneous Dirichlet boundary conditions is given by

$$\begin{aligned} dh(t,x) + hh_x(t,x)dt &= \varepsilon h_{xx}(t,x)dt + G(t,h)\left(\mathbf{m}(\mathbf{x})^{\top}\varphi(h;\Theta)dt + \sigma dW(t)\right) \\ h(t,0) &= h(t,a) = 1.0 \\ h(0,x) &= 0,\ \forall x \in (0,a) \end{aligned} \tag{L.7}$$

where the parameter ε is the viscosity of the medium. Equation (L.7) considers a simple model of a 1D flow of a fluid in a medium with non-zero flow velocities at the two boundaries. The goal is to achieve and maintain a desired flow velocity profile at certain points along the spatial domain. As seen in the desired profile in fig. 5.2e in the main paper, there are 3 areas along the spatial domain with desired flow velocity such that the flow has to be accelerated, then decelerated, and then accelerated again while trying to overcome the stochastic forces

and the dynamics governed by the Burgers SPDE. Similar to the experiments for the Heat SPDE, we consider $\mathbf{m}(\mathbf{x})^\top = [m_1(\mathbf{x}), m_2(\mathbf{x}), m_3(\mathbf{x}), m_4(\mathbf{x})]$ and $G(t,h)$ as an identity operator with the actuators behaving as Gaussian-like exponential functions with the means co-located with the actuator locations at: $\mu = \left[0.2a, 0.3a, 0.5a, 0.7a, 0.8a\right]$ and the spatial effect (variance) of each actuator given by $\sigma_l^2 = (0.1a)^2$, $\forall l = 1, \ldots, 5$. The parameter $a = 1.0\ m$ is the length of the channel along which the fluid is flowing.

This spatial domain was discretized using a grid of 64 points. The numerical scheme used semi-implicit forward Euler discretization for time and central difference approximation for both the 1^{st} and 2^{nd} order derivatives in space. The 1^{st} order derivative terms in the advection term uu_x were evaluated at the current time instant while the 2^{nd} order spatial derivatives in the diffusion term u_{xx} were evaluated at the next time instant, hence the scheme is semi-implicit. Following are values of some other parameters used in our experiments: time discretization $\Delta t = 0.01$, total simulation time = $1.0s$, and the scaling parameter $\kappa = 100$. The cost function considered for the experiments is given by eq. (L.2), where $S := \cup_{i=1}^{3} S_i$ and the indicator function $\mathbb{1}_S(x)$ is given by eq. (L.3) with regions S_1, S_2, S_3 given by eq. (L.4). In addition, $h_{\text{desired}}(t,x) = 2.0\ m/s$ for $x \in S_1 \cup S_3$, which is at the sides, and $h_{\text{desired}}(t,x) = 1.0\ m/s$ for $x \in S_2$, which is in the central region.

The non-linear policy $\varphi(h;\Theta)$ was chosen to be a FNN with 2 hidden layers of 64 neurons each and ReLU activations. The network was trained using the ADAM optimizer for 1000 iterations with 100 trajectories sampled from the Burgers SPDE model per iteration. Each trajectory was 2.0 seconds long with $\Delta t = 0.01$ seconds.

L.4 1D Nagumo SPDE distributed control (Suppression Task)

The stochastic Nagumo equation with Neumann boundary conditions is given by

$$\begin{aligned} dh(t,x) &= \varepsilon h_{xx}(t,x)dt + h(t,x)\big(1-h(t,x)\big)\big(h(t,x)-\alpha\big)dt + G(t,h)\big(\mathbf{m}(\mathbf{x})^\top \varphi(h;\Theta)dt \\ &\quad + \sigma dW(t)\big) \\ h_x(t,0) &= h_x(t,a) = 0 \\ h(0,x) &= \left(1+\exp\Big(-\frac{2-x}{\sqrt{2}}\Big)\right)^{-1} \end{aligned} \tag{L.8}$$

The parameter α determines the speed of a wave traveling down the length of the axon and ε the rate of diffusion. By simulating the deterministic Nagumo equation with $a = 5.0$, $\varepsilon = 1.0$ and $\alpha = -0.5$, we observed that after about 3.5 seconds, the wave completely propagates to the end of the axon. We consider $\mathbf{m}(\mathbf{x})^\top = [m_1(\mathbf{x}), m_2(\mathbf{x}), m_3(\mathbf{x})]$ and $G(t,h)$ as an identity operator with the actuators dynamics $m(\mathbf{x})$ modelled as Gaussian-like exponential functions with actuator centers (mean values) at $\mu = \big[0.7a, 0.8a, 0.9a\big]$ and the spatial effect (variance) of each actuator given by $\sigma_l^2 = (0.1a)^2$, for $l = 1,2,3$. The spatial domain was discretized using a grid of 64 points. The cost function for this experiment is defined as

$$J = \sum_t \sum_x \kappa\big(h_{\text{actual}}(t,x)\big)^2 \cdot \mathbb{1}_S(x)$$

where κ was chosen as 10^{-3}, and the indicator function $\mathbb{1}_S(x)$ is defined as in eq. (L.3) with $S = [0.7a, 0.99a]$. The non-linear policy $\varphi(h;\Theta)$ was chosen to be a FNN with 2 hidden layers of 64 neurons each and ReLU activations. The network was trained using the ADAM optimizer for 1000 iterations with 50 trajectories sampled from the Nagumo SPDE model per iteration. Each trajectory was 3.5 seconds long and $\Delta t = 0.01$ seconds.

APPENDIX M

DERIVATION OF THE LINDBLAD FORM

This section follows the derivation in [171] and adds more detail and intermediate steps for clarity. An open quantum system is a closed quantum system S coupled to another system B, called the environment. The total system can be expressed as $S+B$, which is a closed system. Instead of just including the quantum system S, the closure now includes the system and an environment B which S interacts with. The system S has potentially infinite degrees of freedom and is referred to as the “reduced system", while an environment B with infinite degrees of freedom is referred to as a “reservoir". A reservoir in equilibrium is referred to as a “heat bath" or simply a “bath". The total Hilbert space of $S+B$ is given by

$$\mathscr{H} = \mathscr{H}_S \otimes \mathscr{H}_B \tag{M.1}$$

with time-dependent Hamiltonian

$$\hat{H}(t) = \hat{H}_S \otimes \hat{I}_B + \hat{I}_S \otimes \hat{H}_B + \hat{H}_I(t). \tag{M.2}$$

In real-world applications, properties of B can be unknown and uncontrollable. Therefore, B is often restricted to simple cases for which solutions can be obtained.

Observables of the system S are always realized as operators of the form $\hat{A} \otimes \hat{I}_B$. Therefore, the expectation of the observable $\hat{A}$ of system S is given by

$$\langle \hat{A} \rangle = \mathrm{tr}_S\{A\rho_S\}, \tag{M.3}$$

where tr_S is the partial trace with respect to a complete orthonormal basis in S (i.e the degrees of freedom of S), and $\rho_S = \mathrm{tr}_B\rho$ is the reduced density matrix of S. The reduced density

matrix has evolution given by

$$\rho_S(t) = \mathrm{tr}_B\{\hat{U}(t,0)\rho(0)\hat{U}^\dagger(t,0)\}, \tag{M.4}$$

and the total density matrix is a composition of the system and environment density matrices

$$\rho(t) = \rho_S \otimes \rho_B. \tag{M.5}$$

We can similarly apply the partial trace to the Liouville-von Neumann equation

$$\frac{\mathrm{d}}{\mathrm{d}t}\rho_S = \frac{-i}{\hbar}\mathrm{tr}_B\big[\hat{H}(t), \rho(t)\big]. \tag{M.6}$$

M.1 Dynamical Semigroups

Suppose the open system is initially in an uncorrelated state $\rho(0) = \rho_S \otimes \rho_B$. Then the mapping from initial state to a future state is given by

$$\begin{aligned} \rho_S(0) \to \rho_S(t) &= \hat{V}(t)\rho_S(0) \\ &= \mathrm{tr}_B\{\hat{U}(t,0)[\rho_S(0)\otimes\rho_B]\hat{U}^\dagger(t,0)\}, \end{aligned} \tag{M.7}$$

where $\hat{V}(t)$ is a dynamical mapping operator $\hat{V} : \mathscr{X}(\mathscr{H}_S) \to \mathscr{X}(\mathscr{H}_S)$. $\mathscr{X}(\mathscr{H}_S)$ denotes the space of density matrices of the reduced system S.

Let $\{|\phi_\alpha\rangle\}$ denote a complete orthonormal basis in $\mathscr{H}_B$. The spectral decomposition of the environment is given by

$$\rho_B = \sum_\alpha \lambda_\alpha |\phi_\alpha\rangle\langle\phi_\alpha|, \tag{M.8}$$

where non-negative eigenvalues λ_α satisfy $\sum_\alpha \lambda_\alpha = 1$. Using this decomposition, eq. (M.7)

can be written as

$$\hat{V}(t)\rho_S(0) = \mathrm{tr}_B\{\hat{U}(t,0)[\rho_S(0)\otimes\rho_B]\hat{U}^\dagger(t,0)\} \tag{M.9}$$

$$= \sum_\beta \langle\phi_\beta|\hat{U}(t,0)[\rho_S(0)\otimes\rho_B]\hat{U}^\dagger(t,0)|\phi_\beta\rangle \tag{M.10}$$

$$= \sum_\beta \langle\phi_\beta|\hat{U}(t,0)[\rho_S(0)\otimes\sum_\alpha \lambda_\alpha|\phi_\alpha\rangle\langle\phi_\alpha|]\hat{U}^\dagger(t,0)|\phi_\beta\rangle \tag{M.11}$$

$$= \sum_{\alpha,\beta} \lambda_\alpha\langle\phi_\beta|\hat{U}(t,0)|\phi_\alpha\rangle\rho_S(0)\langle\phi_\alpha|\hat{U}^\dagger(t,0)|\phi_\beta\rangle \tag{M.12}$$

$$= \sum_{\alpha,\beta} \lambda_\beta\langle\phi_\alpha|\hat{U}(t,0)|\phi_\beta\rangle\rho_S(0)(\langle\phi_\alpha|\hat{U}(t,0)|\phi_\beta\rangle)^\dagger \tag{M.13}$$

$$= \sum_{\alpha,\beta} \hat{W}_{\alpha\beta}(t)\rho_S(0)\hat{W}^\dagger_{\alpha\beta}(t), \tag{M.14}$$

where the operators $\hat{W}_{\alpha\beta}(t)$ in $\mathscr{H}_S$ are defined by

$$\hat{W}_{\alpha\beta}(t) \equiv \sqrt{\lambda_\beta}\langle\phi_\alpha|\hat{U}(t,0)|\phi_\beta\rangle, \tag{M.15}$$

and satisfies the condition

$$\sum_{\alpha\beta} \hat{W}^\dagger_{\alpha\beta}(t)\hat{W}_{\alpha\beta}(t) = \hat{I}_S. \tag{M.16}$$

Finally, note that the dynamical map operator is trace-preserving

$$\mathrm{tr}_S\{\hat{V}(t)\rho_S\} = \mathrm{tr}_S\Big\{\sum_{\alpha,\beta} \hat{W}_{\alpha\beta}(t)\rho_S(0)\hat{W}^\dagger_{\alpha\beta}(t)\Big\} \tag{M.17}$$

$$= \mathrm{tr}_S\Big\{\sum_{\alpha,\beta} \hat{W}^\dagger_{\alpha\beta}(t)\hat{W}_{\alpha\beta}(t)\rho_S(0)\Big\} \tag{M.18}$$

$$= \mathrm{tr}_S\rho_S(0) \tag{M.19}$$

$$= 1. \tag{M.20}$$

Thus, we conclude that the dynamical map operator $\hat{V}(t)$ is a convex-linear, positive, trace-preserving operator.

The dynamical map operator $\hat{V}(t)$ defined above is defined for fixed time t. Allowing t to vary produces a one-parameter family $\{V(t)|t \geq 0\}$ of dynamical maps which describe the whole future time evolution of the system. This time evolution can be very involved. However, this can be treated by applying a Markovian assumption.

M.2 Markovian Quantum Master Equation

If the characteristic timescales over which the reservoir correlation functions decay are much smaller than the characteristic timescale of the system, then we can neglect memory effects and apply a Markovian assumption. For a homogeneous process (a process whose propagator only depends on the difference of time arguments) the dynamical map is given by

$$\hat{V}(t_1)\hat{V}(t_2) = \hat{V}(t_1+t_2), \quad t_1, t_2 \geq 0 \tag{M.21}$$

for which there exists a linear operator $\hat{\mathscr{L}}$ which is the generator of the map. The dynamical map operator can then be written as

$$\hat{V}(t) = \exp(\hat{\mathscr{L}}t) \tag{M.22}$$

which yields the “Markovian Quantum Master Equation" for the density matrix

$$\frac{\mathrm{d}}{\mathrm{d}t}\rho_S(t) = \hat{\mathscr{L}}\rho_S(t) \tag{M.23}$$

In order to construct the most general form of $\hat{\mathscr{L}}$, consider first the simple case of a finite dimensional Hilbert space with $dim\mathscr{H}_S = N$. The corresponding Liouville space (direct product of Hilbert spaces) is of dimension N^2. Define the Hilbert-Schmidt inner product as

$$(\hat{A}, \hat{B}) \equiv \mathrm{tr}\{\hat{A}^\dagger \hat{B}\}. \tag{M.24}$$

We Choose a complete set of orthonormal operators $\hat{F}_i$, $i = 1,2,\ldots,N^2$ such that

$$(\hat{F}_i, \hat{F}_j) = \mathrm{tr}_S\{\hat{F}_i^\dagger \hat{F}_j\} = \delta_{ij}. \tag{M.25}$$

For convenience, set $\hat{F}_{N^2} = (\frac{1}{N})^{1/2}\hat{I}_S$ such that all the other basis operators are traceless (i.e. $\mathrm{tr}_S\hat{F}_i = 0$, $\forall\ i = 1,2,\ldots,N^2-1$). Now, applying the completeness relation, we can write eq. (M.15) in terms of our basis operators

$$\hat{W}_{\alpha\beta}(t) = \sum_{i=1}^{N^2} \hat{F}_i(\hat{F}_i, \hat{W}_{\alpha\beta}(t)) \tag{M.26}$$

The dynamical map operator can then be written as

$$\begin{aligned}
\hat{V}(t)\rho_S(t) &= \sum_{\alpha,\beta} \hat{W}_{\alpha\beta}(t)\rho_S(0)\hat{W}_{\alpha\beta}^\dagger(t) \\
&= \sum_{i,j=1}^{N^2} \hat{F}_i(\hat{F}_i, \hat{W}_{\alpha\beta}(t))\rho_S(0)\hat{F}_j^\dagger(\hat{F}_i, \hat{W}_{\alpha\beta}(t))^* \\
&= \sum_{i,j=1}^{N^2} (\hat{F}_i, \hat{W}_{\alpha\beta}(t))(\hat{F}_i, \hat{W}_{\alpha\beta}(t))^*\hat{F}_i\rho_S(0)\hat{F}_j^\dagger \\
&= \sum_{i,j=1}^{N^2} c_{ij}\hat{F}_i\rho_S(0)\hat{F}_j^\dagger,
\end{aligned} \tag{M.27}$$

where $c_{ij} = (\hat{F}_i, \hat{W}_{\alpha\beta}(t))(\hat{F}_i, \hat{W}_{\alpha\beta}(t))^*$, and note that the matrix c is Hermitian and positive semidefinite.

Now, with eq. (M.22) and eq. (M.23), the generator is given by

$$\begin{aligned}
\hat{\mathscr{L}}\rho_S &= \frac{\ln \hat{V}(t)\rho_S}{t} \\
&\approx \lim_{\varepsilon\to 0}\frac{1}{\varepsilon}\left[V(\varepsilon)\rho_S - \rho_S\right] \\
&= \lim_{\varepsilon\to 0}\frac{1}{\varepsilon}\left[\sum_{i,j=1}^{N^2} c_{ij}\hat{F}_i\rho_S\hat{F}_j^\dagger - \rho_S\right] \\
&= \lim_{\varepsilon\to 0}\frac{1}{\varepsilon}\left[\frac{1}{N}(C_{N^2N^2}(\varepsilon)-N)\rho_S + \frac{1}{\sqrt{N}}\sum_{i=1}^{N^2-1}(C_{iN^2}(\varepsilon)\hat{F}_i\rho_S C_{N^2i}\rho_S\hat{F}_i^\dagger \right. \\
&\qquad \left. + \sum_{i,j=1}^{N^2-1} C_{ij}(\varepsilon)\hat{F}_i\rho_S\hat{F}_j^\dagger\right]. \qquad \text{(M.28)}
\end{aligned}$$

Now define constants and operators

$$\begin{aligned}
a_{N^2N^2} &\equiv \lim_{\varepsilon\to 0}\frac{c_{N^2N^2}(\varepsilon)-N}{\varepsilon} & \hat{F} &\equiv \frac{1}{\sqrt{N}}\sum_{i=1}^{N^2-1} a_{iN^2}(\varepsilon)\hat{F}_i \\
a_{iN^2} &\equiv \lim_{\varepsilon\to 0}\frac{c_{iN^2}(\varepsilon)}{\varepsilon}, \quad i=1,\ldots,N^2-1 & \hat{G} &\equiv \frac{1}{2N}a_{N^2N^2}\hat{I}_S + \frac{1}{2}(\hat{F}^\dagger + \hat{F}) \\
a_{ij} &\equiv \lim_{\varepsilon\to 0}\frac{c_{ij}(\varepsilon)}{\varepsilon}, \quad i=1,\ldots,N^2-1 & \hat{H} &\equiv \frac{1}{2i}(\hat{F}^\dagger + \hat{F}).
\end{aligned}$$

Applying these to eq. (M.28), we obtain

$$\begin{aligned}
\hat{\mathscr{L}}\rho_S &= \frac{1}{N}a_{N^2N^2}\rho_S + \hat{F}\rho_S + \rho_S\hat{F}^\dagger + \sum_{i,j=1}^{N^2-1} a_{ij}\hat{F}_i\rho_S\hat{F}_j^\dagger \\
&= \frac{1}{2N}a_{N^2N^2}(\hat{I}_S\rho_S + \rho_S\hat{I}_S) + \frac{1}{2}\hat{F}\rho_S + \frac{1}{2}\hat{F}\rho_S + \frac{1}{2}\rho_S\hat{F}^\dagger + \frac{1}{2}\rho_S\hat{F}^\dagger + \sum_{i,j=1}^{N^2-1} a_{ij}\hat{F}_i\rho_S\hat{F}_j^\dagger \\
&= \hat{G}\rho_S + \rho_S\hat{G}^\dagger - \frac{1}{2}\rho_S\hat{F} - \frac{1}{2}\hat{F}^\dagger\rho_S + \frac{1}{2}\hat{F}\rho_S + \frac{1}{2}\rho_S\hat{F}^\dagger + \sum_{i,j=1}^{N^2-1} a_{ij}\hat{F}_i\rho_S\hat{F}_j^\dagger \\
&= \{\hat{G},\rho_S\} + \frac{1}{2}(\hat{F}-\hat{F}^\dagger)\rho_S + \frac{1}{2}\rho_S(\hat{F}^\dagger - \hat{F}) + \sum_{i,j=1}^{N^2-1} a_{ij}\hat{F}_i\rho_S\hat{F}_j^\dagger \\
&= \{\hat{G},\rho_S\} - i[\hat{H},\rho_S] + \sum_{i,j=1}^{N^2-1} a_{ij}\hat{F}_i\rho_S\hat{F}_j^\dagger. \qquad \text{(M.29)}
\end{aligned}$$

Note that the semigroup $\hat{\mathscr{L}}$ is trace preserving:

$$\begin{aligned}
0 = \mathrm{tr}_S\{\hat{\mathscr{L}}\rho_S\} &= \mathrm{tr}_S\left\{\hat{G}\rho_S + \rho_S\hat{G}^\dagger + \frac{1}{2}(\hat{F}-\hat{F}^\dagger)\rho_S + \frac{1}{2}\rho_S(\hat{F}^\dagger-\hat{F}) + \sum_{i,j=1}^{N^2-1} a_{ij}\hat{F}_i\rho_S\hat{F}_j^\dagger\right\} \\
&= \mathrm{tr}_S\left\{\left(2\hat{G} + \frac{1}{2}(\hat{F}-\hat{F}^\dagger) + \frac{1}{2}(\hat{F}^\dagger-\hat{F}) + \sum_{i,j=1}^{N^2-1} a_{ij}\hat{F}_j\hat{F}_i^\dagger\right)\rho_S\right\} \\
&= \mathrm{tr}_S\left\{\left(2\hat{G} + \sum_{i,j=1}^{N^2-1} a_{ij}\hat{F}_j\hat{F}_i^\dagger\right)\rho_S\right\},
\end{aligned} \tag{M.30}$$

which implies that

$$\hat{G} = -\frac{1}{2}\sum_{i,j=1}^{N^2-1} a_{ij}\hat{F}_j\hat{F}_i^\dagger. \tag{M.31}$$

Substituting in $\hat{G}$ yields the “standard form" of the generator

$$\hat{\mathscr{L}}\rho_S = -i[\hat{H},\rho_S] + \sum_{i,j=1}^{N^2-1} a_{ij}\left(\hat{F}_i\rho_S\hat{F}_j^\dagger - \frac{1}{2}\{\hat{F}_j^\dagger\hat{F}_i,\rho_S\}\right). \tag{M.32}$$

Now, since a is a positive-definite matrix, it can be diagonalized via the Schur decomposition

$$uau^\dagger = \mathrm{diag}(\gamma_1,\ldots,\gamma_{N^2-1}) \tag{M.33}$$

where γ_i are non-negative eigenvalues. By this decomposition, $\hat{F}_i$ can be expressed as

$$\hat{F}_i = \sum_{k=1}^{N^2-1} u_{ki}\hat{A}_k \tag{M.34}$$

Finally substituting in the diagonalization into eq. (M.32) produces the diagonal form of the generator, which is also called the “Lindblad form"

$$\hat{\mathscr{L}}\rho_S = -i[\hat{H},\rho_S] + \sum_{k=1}^{N^2-1}\gamma_k\left(\hat{A}_k\rho_S\hat{A}_k^\dagger - \frac{1}{2}\hat{A}_k^\dagger\hat{A}_k\rho_S - \frac{1}{2}\rho_S\hat{A}_k^\dagger\hat{A}_k\right). \tag{M.35}$$

The commutator term describes a unitary part of the evolution generated by the Hamiltonian $\hat{H}$, and the γ_k eigenvalues are given in terms of correlation functions of the environment and

describe relaxation rates for the different decay modes of the system. The operators $\hat{A}_k$ are linear combinations of the basis operators $\hat{F}_i$ and are called “Lindblad operators".

APPENDIX N

DERIVATION OF THE BELAVKIN EQUATION FOR DISCRETE QND MEASUREMENT

This derivation follows that of [171] and adds more detail and intermediate steps for clarity. The Belavkin equation is a stochastic differential equation used for non-demolition measurement of a quantum system, which can be used for filtering and feedback control. The aparatus used for measurement is set up in such a way that we can ignore memory effects in the dynamics which would arise due to the coupled dynamics, thus implying that these system-environment interactions are “slow" in comparison to the system dynamics. We assume that in this case we have Markovian dynamics (i.e. the state at the current time only depends on the state at the previous time and not states at time steps before the previous time). The Markovian evolution is given by a continuous semigroup $\{T_t\}_{t\geq 0}$ of completely positive maps. The Lindblad master equation (derived separately) describes the generator of T_t and is given by

$$\frac{d}{dt}|_{t=0} = L(\rho) = -i[H,\rho] + i[V+V^*,\rho] - \frac{1}{2}\{V^*V,\rho\} + V\rho V^*, \tag{N.1}$$

where H is a Hamiltonian that describes the evolution of the closed system, and V captures interactions with the environment.

In order to arrive at the continuous measurement Belavkin master equation, we first start by considering a photon counting experiment. In this experiment, we couple a laser to an atom in a “forward channel" and split the laser to “side channels", where we measure the emission of photons due to excitation of the atom by the laser. In this case, we can “unravel"

this master equation by writing it as

$$L(\rho) = \mathscr{L}(\rho) + \mathscr{J}(\rho) = -i[H,\rho] + i[V+V^*,\rho] - \frac{1}{2}\{V^*V,\rho\} + (1-|\kappa_s|^2)V\rho V^* + |\kappa_s|^2 V\rho V^* \tag{N.2}$$

where $\mathscr{J}(\rho) = |\kappa_s|^2 V\rho V^*$ and $\mathscr{L}(\rho)$ contains the remaining terms. The term $|\kappa_s|^2$ is the decay rate of photons into the side channel, and we will denote remaining photons in the forward channel as $|\kappa_f|^2$, such that $|\kappa_s|^2 + |\kappa_f|^2 = 1$.

In order to determine the measure space for the photon counting measurement, we first consider a single measurement outcome over an arbitrary finite time interval $[0,t)$. Photons are not detected at every time instant, however over the experiment specifies those time instants for which a photon is measured. Over the set $[0,t)$, the measurement outcome is therefore those time instants for which a photon is measured. For example consider an specific outcome that measures k photons on $[0,t)$. The set of times would be $\{t_1,t_2,\ldots,t_k\}$. Since k is arbitrary for each experiment, the space of outcomes is

$$\Omega([0,t)) := \cup_{n=0}^{\infty}\{\sigma \subset [0,t) : |\sigma| = n\} = \cup_{n=0}^{\infty}\Omega_n([0,t)) \tag{N.3}$$

In other words, the sample space is the union of all possible outcomes, each outcome being a number of counted photons associated with a set of counting times. With this we are able to form a sample space.

Consider the space of n-tuples $[0,t)^n$ with Borel σ-algebra and measure $\frac{1}{n!}\lambda_n$, where λ_n is the Lebesgue measure. Then the “counting current"

$$j_n : (t_1,\ldots,t_n) \in [0,t)^n \to \{t_1,\ldots,t_n\} \in \Omega_n([0,t)) \tag{N.4}$$

induces a σ-algebra $\Sigma_n([0,t))$ and measure μ_n on $\Omega_n([0,t))$. For a given experiment, each continuum of time produces a set of times for which a photon is counted. Define a measure

μ on $\Omega([0,t))$ such that $\mu = \mu_n$ on $\Omega_n([0,t))$ and $\mu(\{\phi\}) = 1$. Note that μ_n is the measure over each experiment, while μ is the measure for the sample space.

Next, we wish to find an expression for the unnormalized state. First define the evolution operator for experiment outcome $\omega = \{t_1, \ldots, t_k\} \in \Omega^t([0,t))$ with ordered times $0 \leq t_1 \leq \cdots \leq t_k < t$ as

$$W_t(\omega)(\rho) := \exp((t-t_k)\mathscr{L})\,\mathscr{J} \ldots \mathscr{J} \exp((t_2-t_1)\mathscr{L})\,\mathscr{J} \exp(t_1\mathscr{L})(\rho) \tag{N.5}$$

Next, denote the unnormalized state of the two-level atom at time t with initial state ρ conditioned on the outcome of the experiment being in a set $E \in \Sigma^t$ as $\mathscr{M}^t[E](\rho)$. Davies first showed that it is given by

$$\mathscr{M}^t[E](\rho) = \int_E W_t(\omega)(\rho) d\mu(\omega). \tag{N.6}$$

With this, we can define the probability that event E occurs if the initial state is ρ as $\mathbb{P}^t_\rho[E] := \mathrm{tr}(\mathscr{M}^t[E](\rho))$. This probability measure forms a consistent family of probability measures $\{\mathbb{P}^t_\rho\}_{t\geq 0}$ across time, i.e. $\mathbb{P}^{t+s}_\rho[E] = \mathbb{P}^t_\rho[E]$, $\forall E \in \Sigma^t, s \geq 0$. We use Kolmogorov's extension theorem to extend this to a single probability measure $\mathbb{P}_\rho$ on the σ-algebra Σ^∞ of the sample space Ω^∞. These denote the probability measure, σ-algebra, and sample space over all possible outcomes, respectively.

We define a random variable N_t on the measure space $(\Omega^\infty, \Sigma^\infty, \mathbb{P}_\rho)$ that takes events $E \in \Omega^\infty$ and counts photons

$$N_t := \Omega^\infty \to \mathbb{N} : \omega \mapsto |\omega \cap [0,t)| \tag{N.7}$$

This random variable N_t counts the number of photon counting times that appear in experiment outcome ω. It has differential form $\mathrm{d}N_t = N_{t+\mathrm{d}t} - N_t$ such that if the current time appears in the outcome $t \in \omega$ (a photon is measured), the count is incremented $\mathrm{d}N_t(\omega) = 1$.

Otherwise, no photon is measured and the count is not incremented $\mathrm{d}N_t(\omega) = 0$. The counting process has Itô rules $\mathrm{d}N_t \mathrm{d}N_t = \mathrm{d}N_t$ since $\mathrm{d}N_t \leq 1$, and $\mathrm{d}N_t \mathrm{d}t = 0$. As a result of the random variable $\mathrm{d}N_t$, the evolution of the density is stochastic, as represented by the 2x2 matrix random variable $\{\rho_\bullet^t\}_{t\geq 0}$ that takes events $E \in \Omega^\infty$ and produces a 2x2 matrix $\rho_\omega^t \in M_2$

$$\rho_\bullet^t : \Omega^\infty \to M_2 : \omega \mapsto \rho_\omega^t \tag{N.8}$$

where for realization ω, ρ_ω^t is defined as

$$\rho_\omega^t := \frac{W_t(\omega \cap [0,t))(\rho)}{\mathrm{tr}\{W_t(\omega \cap [0,t))(\rho)\}} \tag{N.9}$$

Here, $W_t(\omega \cap [0,t))(\rho)$ takes as input the photon measurement times that are inside the relevant time window. We explicitly take the intersection here because Ω^∞ is no longer time indexed, however W_t has the same form as above.

Next, we relate the two stochastic processes through the differential equation

$$\mathrm{d}\rho_\bullet^t = \alpha_t \mathrm{d}t + \beta_t \mathrm{d}N_t, \tag{N.10}$$

where α_t and β_t are processes that can be determined by differentiating eq. (N.9). This differential equation is split into two parts: one were $\mathrm{d}t$ dominates, i.e. $t \notin \omega$, and one where $\mathrm{d}N_t$ dominates, i.e. $t \in \omega$. Let us first examine the second part.

When $t \in \omega$, as noted above $\mathrm{d}N_t(\omega) = 1$, which dominates $\mathrm{d}t$, i.e. $\mathrm{d}N_t \gg \mathrm{d}t$. In this case, we have

$$\mathrm{d}\rho_\bullet^t = \beta_t \mathrm{d}N_t = \beta_t \tag{N.11}$$

Splitting up $\mathrm{d}\rho_\bullet^t$ and solving for β yields

$$\begin{aligned}\beta_t &= \rho_\omega^{t+\mathrm{d}t} - \rho_\omega^t \\ &= \frac{W_{t+\mathrm{d}t}(\omega \cap [0,t+\mathrm{d}t))(\rho)}{\mathrm{tr}\{W_{t+\mathrm{d}t}(\omega \cap [0,t+\mathrm{d}t))(\rho)\}} - \rho_\omega^t \\ &= \frac{\exp((t+\mathrm{d}t-t)\mathscr{L})\,\mathscr{J}(\rho_\omega^t)}{\mathrm{tr}\{\exp((t+\mathrm{d}t-t)\mathscr{L})\,\mathscr{J}(\rho_\omega^t)\}} - \rho_\omega^t \\ &= \frac{\mathscr{J}(\rho_\omega^t)}{\mathrm{tr}\{\mathscr{J}(\rho_\omega^t)\}} - \rho_\omega^t \end{aligned} \tag{N.12}$$

On the other hand, when $t \notin \omega$, then $\mathrm{d}N_t(\omega) = 0$, so $\mathrm{d}t$ dominates $\mathrm{d}N_t(\omega)$, i.e. $\mathrm{d}t \gg \mathrm{d}N_t$. In this case, we have

$$\mathrm{d}\rho_\bullet^t = \alpha \mathrm{d}t \tag{N.13}$$

solving for α yields

$$\begin{aligned}\alpha &= \frac{\mathrm{d}}{\mathrm{d}s}\bigg|_{s=t} \rho_\omega^t \\ &= \frac{\mathrm{d}}{\mathrm{d}s}\bigg|_{s=t} \frac{e^{(s-t)\mathscr{L}}\,\mathscr{J}(\rho_\omega^t)}{\mathrm{tr}\{e^{(s-t)\mathscr{L}}\,\mathscr{J}(\rho_\omega^t)\}} \end{aligned} \tag{N.14}$$

However, $\mathscr{J}(\rho_\omega^t) = |\kappa_s|^2 V \rho_\omega^t V^*$ is neglected since $\mathrm{d}t \gg \mathrm{d}N_t$, i.e. the system evolution dominates system-environment interaction so that

$$\rho_\omega^t = \frac{e^{(s-t)\mathscr{L}}}{\mathrm{tr}\{e^{(s-t)\mathscr{L}}\}} \tag{N.15}$$

evaluating the derivative in eq. (N.14) via quotient rule yields

$$\begin{aligned}\alpha &= \frac{\mathscr{L}e^{(s-t)\mathscr{L}}\mathrm{tr}\{e^{(s-t)\mathscr{L}}\} - e^{(s-t)\mathscr{L}}\mathrm{tr}\{\mathscr{L}e^{(s-t)\mathscr{L}}\}}{\mathrm{tr}\{e^{(s-t)\mathscr{L}}\}^2}\bigg|_{s=t} \\ &= \frac{\mathscr{L}e^{(s-t)\mathscr{L}}\mathrm{tr}\{e^{(s-t)\mathscr{L}}\}}{\mathrm{tr}\{e^{(s-t)\mathscr{L}}\}^2}\bigg|_{s=t} - \frac{e^{(s-t)\mathscr{L}}\mathrm{tr}\{\mathscr{L}e^{(s-t)\mathscr{L}}\}}{\mathrm{tr}\{e^{(s-t)\mathscr{L}}\}^2}\bigg|_{s=t} \end{aligned} \tag{N.16}$$

evaluating the first term for $s = t$ and splitting the trace in the numerator of the second term

yields

$$\alpha = \mathscr{L} - \frac{e^{(s-t)\mathscr{L}}\mathrm{tr}\{e^{(s-t)\mathscr{L}}\}\mathrm{tr}\{\mathscr{L}\}}{\mathrm{tr}\{e^{(s-t)\mathscr{L}}\}^2}\bigg|_{s=t} \tag{N.17}$$

also, using eq. (N.15), we obtain

$$\begin{aligned}\alpha &= \mathscr{L} - \frac{\rho_\omega^t \mathrm{tr}\{e^{(s-t)\mathscr{L}}\}\mathrm{tr}\{\mathscr{L}\}}{\mathrm{tr}\{e^{(s-t)\mathscr{L}}\}} \\ &= \mathscr{L} - \rho_\omega^t \mathrm{tr}\{\mathscr{L}\}. \end{aligned} \tag{N.18}$$

Finally, noting that $\mathrm{tr}\{L\} = 0 = \mathrm{tr}\{\mathscr{L} + \mathscr{J}\}$, we have that $\mathrm{tr}\{\mathscr{L}\} = -\mathrm{tr}\{\mathscr{J}\}$. This results in

$$\alpha = \mathscr{L}(\rho_\omega^t) + \mathrm{tr}\{\mathscr{J}(\rho_\omega^t\}\rho_\omega^t \tag{N.19}$$

The Belavkin master equation for the counting process results from plugging in eq. (N.19) and eq. (N.12) into eq. (N.10), yielding

$$\begin{aligned}\mathrm{d}\rho_\bullet^t &= \left[\mathscr{L} + \rho_\bullet^t \mathrm{tr}\{\mathscr{J}(\rho_\bullet^t)\}\right]\mathrm{d}t + \left[\frac{\mathscr{J}(\rho_\bullet^t}{\mathrm{tr}\{\mathscr{J}(\rho_\bullet^t)\}} - \rho_\bullet^t\right]\mathrm{d}N_t \\ &= L(\rho_\bullet^t)\mathrm{d}t + \mathscr{J}(\rho_\bullet^t)\mathrm{d}t + \rho_\bullet^t \mathrm{tr}\{\mathscr{J}(\rho_\bullet^t)\}\mathrm{d}t + \frac{\mathscr{J}(\rho_\bullet^t)}{\mathrm{tr}\{\mathscr{J}(\rho_\bullet^t\}}\mathrm{d}N_t - \rho_\bullet^t \mathrm{d}N_t \\ &= L(\rho_\bullet^t)\mathrm{d}t + \frac{\mathscr{J}(\rho_\bullet^t)}{\mathrm{tr}\{\mathscr{J}(\rho_\bullet^t)\}}\mathrm{tr}\{\mathscr{J}(\rho_\bullet^t)\}\mathrm{d}t + \rho_\bullet^t \mathrm{tr}\{\mathscr{J}(\rho_\bullet^t)\}\mathrm{d}t + \frac{\mathscr{J}(\rho_\bullet^t)}{\mathrm{tr}\{\mathscr{J}(\rho_\bullet^t\}}\mathrm{d}N_t - \rho_\bullet^t \mathrm{d}N_t \\ &= L(\rho_\bullet^t)\mathrm{d}t + \frac{\mathscr{J}(\rho_\bullet^t)}{\mathrm{tr}\{\mathscr{J}(\rho_\bullet^t)\}}\Big(\mathrm{d}N_t - \mathrm{tr}\{\mathscr{J}(\rho_\bullet^t)\}\mathrm{d}t\Big) - \rho_\bullet^t\Big(\mathrm{d}N_t - \mathrm{tr}\{\mathscr{J}(\rho_\bullet^t)\}\mathrm{d}t\Big) \\ &= L(\rho_\bullet^t)\mathrm{d}t + \left(\frac{\mathscr{J}(\rho_\bullet^t)}{\mathrm{tr}\{\mathscr{J}(\rho_\bullet^t)\}} - \rho_\bullet^t\right)\Big(\mathrm{d}N_t - \mathrm{tr}\{\mathscr{J}(\rho_\bullet^t)\}\mathrm{d}t\Big) \end{aligned} \tag{N.20}$$

Defining the so called Innovating Martingale as

$$\mathrm{d}M_t := \mathrm{d}N_t - \mathrm{tr}\{\mathscr{J}(\rho_\bullet^t)\} \tag{N.21}$$

with $M_0 = 0$ yields the compact form of the Belavkin equation

$$\mathrm{d}\rho_\bullet^t = L(\rho_\bullet^t) + \Big(\frac{\mathscr{J}(\rho_\bullet^t)}{\mathrm{tr}\{\mathscr{J}(\rho_\bullet^t)\}} - \rho_\bullet^t\Big)\mathrm{d}M_t \tag{N.22}$$

APPENDIX O

CHANGE OF MEASURE FOR CONTROLLED QND OPEN QUANTUM SYSTEMS

Under our current notation, we have the controlled and uncontrolled processes

$$\mathrm{d}\rho_t = F(\rho_t)\mathrm{d}t + B(\rho_t)\mathrm{d}W_t \tag{O.1}$$

$$\mathrm{d}\tilde{\rho}_t = F(\tilde{\rho}_t)\mathrm{d}t - i\mathbf{u}^\top \left[H_{\mathbf{u}}, \tilde{\rho}_t\right]\mathrm{d}t + B(\tilde{\rho}_t)\mathrm{d}\tilde{W}_t \tag{O.2}$$

which, under standard assumptions, admits unique weak solutions of the form [194]

$$\rho_t = \rho_0 + \int_0^t F(\rho_s)\mathrm{d}s + \int_0^t B(\rho_s)\mathrm{d}W_s \tag{O.3}$$

$$\tilde{\rho}_t = \rho_0 + \int_0^t F(\tilde{\rho}_s)\mathrm{d}s - i\int_0^t \mathbf{u}^\top \left[H_{\mathbf{u}}, \tilde{\rho}_s\right]\mathrm{d}s + \int_0^t B(\tilde{\rho}_s)\mathrm{d}\tilde{W}_s. \tag{O.4}$$

Next, we assume that $B(\rho)$ is invertible $\forall \rho \in \mathscr{H}_S$. In order to realize the RN derivative or change of measures between the path measure over uncontrolled trajectories and the path measure over controlled trajectories, we write the uncontrolled solution in terms of the controlled diffusion

$$\rho_t = \rho_0 + \int_0^t F(\rho_s)\mathrm{d}s + \int_0^t B(\rho_s)\mathrm{d}\tilde{W}_s + \int_0^t B(\rho_s)\mathscr{P}(s)\mathrm{d}s, \tag{O.5}$$

where $\mathscr{P}(s)$ is a cádlág process (right continuous everywhere) that represents how the diffusions are related, and is given by

$$\mathscr{P}(s) = -iB(\rho_s)^{-1}\mathbf{u}^\top \left[H_{\mathbf{u}}, \rho_s\right] \tag{O.6}$$

With this, we can apply a form of Girsanov's theorem with

$$\tilde{W}_t = W_t - \int_0^t \mathscr{P}(s)\mathrm{d}s \tag{O.7}$$

Thus we arrive at the change of measure

$$\begin{aligned}\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}} &= \exp\Big(-\int_0^T \langle \mathscr{P}(s), \mathrm{d}W_s\rangle - \frac{1}{2}\int_0^T ||\mathscr{P}(s)||^2\mathrm{d}s\Big) \\ &= \exp\Bigg(-\int_0^T \Big\langle -iB(\rho_s)^{-1}\mathbf{u}^\top\Big[H_\mathbf{u},\rho_s\Big], \mathrm{d}W_s\Big\rangle \\ &\qquad -\frac{1}{2}\int_0^T \Big\langle -iB(\rho_s)^{-1}\mathbf{u}^\top\Big[H_\mathbf{u},\rho_s\Big], -iB(\rho_s)^{-1}\mathbf{u}^\top\Big[H_\mathbf{u},\rho_s\Big]\Big\rangle \mathrm{d}s\Bigg)\end{aligned} \tag{O.8}$$

Note that the inner product in this Hilbert space is given by the Hilbert-Schmidt inner product $\langle A,B\rangle = \mathrm{Tr}[A^\dagger B]$, yielding

$$\begin{aligned}\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}} = \exp\Bigg\{ &-\int_0^T \mathrm{Tr}\Big(-iB(\rho_s)^{-1}\mathbf{u}^\top\Big[H_\mathbf{u},\rho_s\Big]\Big)^\dagger \mathrm{d}W_s \\ &-\frac{1}{2}\int_0^T \mathrm{Tr}\Big(-iB(\rho_s)^{-1}\mathbf{u}^\top\Big[H_\mathbf{u},\rho_s\Big]\Big)^\dagger\Big(-iB(\rho_s)^{-1}\Big[\sum_j u_{t,j}H_j,\rho_s\Big]\Big)\mathrm{d}s\Bigg\}\end{aligned} \tag{O.9}$$

However, since H_j and ρ_s are Hermitian, we can write

$$\begin{aligned}\Big(\mathbf{u}^\top\Big[H_\mathbf{u},\rho_s\Big]\Big)^\dagger &= \Big(\sum_j u_{t,j}H_j\rho_s - \rho_s\sum_j u_{t,j}H_j\Big)^\dagger \\ &= \Big(\rho_s\sum_j u_{t,j}H_j - \sum_j u_{t,j}H_j\rho_s\Big) \\ &= -\mathbf{u}^\top\Big[H_\mathbf{u},\rho_s\Big]\end{aligned} \tag{O.10}$$

Also, $B(\rho_s)$ is Hermitian since it is composed of an operator with its Hermitian conjugate

and a trace which is similarly Hermitian. Thus we obtain the final form

$$\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}(\mathbf{u})} = \exp\left(\int_0^T \mathbf{u}^\top \mathrm{Tr}_\rho \left\{ i\big[\mathbf{H}_u, \rho_s\big] B(\rho_s)^{-1} \mathrm{d}W_s \right\} \right.$$
$$\left. + \frac{1}{2} \int_0^T \mathbf{u}^\top \mathrm{Tr}_\rho \left\{ \big[\mathbf{H}_u, \rho_s\big] B(\rho_s)^{-2} \big[\mathbf{H}_u^\top, \rho_s\big] \right\} \mathbf{u} \, \mathrm{d}s \right). \quad \text{(O.11)}$$

APPENDIX P

DERIVATION OF THE VARIATIONAL OPTIMIZATION-BASED FEEDBACK CONTROLLER FOR QND MEASUREMENT OF OPEN QUANTUM SYSTEMS

The following components were provided in earlier sections and repeated here for clarity:

P.1 Controlled and Uncontrolled Dynamics

$$\mathscr{L}: \quad \mathrm{d}\rho_t = F(\rho_t)\mathrm{d}t + B(\rho_t)\mathrm{d}W_t \tag{P.1}$$

$$\tilde{\mathscr{L}}(\mathbf{u}): \quad \mathrm{d}\tilde{\rho}_t = F(\tilde{\rho}_t)\mathrm{d}t - i\sum_j u_{t,j}\big[H_{u,j},\tilde{\rho}_t\big]\mathrm{d}t + B(\tilde{\rho}_t)\mathrm{d}\tilde{W}_t \tag{P.2}$$

P.2 Change of Measure

$$\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}(\mathbf{u})} = \exp\left(\int_0^T \mathbf{u}^\top \mathrm{Tr}_\rho\Big\{i\big[\mathbf{H}_u,\rho_s\big]B(\rho_s)^{-1}\mathrm{d}W_s\Big\} + \frac{1}{2}\int_0^T \mathbf{u}^\top \mathrm{Tr}_\rho\Big\{\big[\mathbf{H}_u,\rho_s\big]B(\rho_s)^{-2}\big[\mathbf{H}_u^\top,\rho_s\big]\Big\}\mathbf{u}\,\mathrm{d}s\right), \tag{P.3}$$

where transposes are defined relative to the control degrees of freedom and the trace is defined over the degrees of freedom of the density.

P.3 Legendre Transformation and Optimal Measure

$$-\frac{1}{r}\log\mathbb{E}_{\mathscr{L}}\Big[\exp(-rJ)\Big] = \min_{\mathscr{U}(\cdot,\cdot)}\Big[\mathbb{E}_{\tilde{\mathscr{L}}}(J) + \frac{1}{r}S_c(\tilde{\mathscr{L}}\,\big|\big|\,\mathscr{L})\Big], \tag{P.4}$$

where for absolutely continuous measures $\tilde{\mathscr{L}} >> \mathscr{L}$ is defined as

$$S_c(\tilde{\mathscr{L}}||\mathscr{L}) := \int_\Omega \ln\frac{\mathrm{d}\tilde{\mathscr{L}}(\mathbf{u})}{\mathrm{d}\mathscr{L}}\mathrm{d}\tilde{\mathscr{L}}(\mathbf{u}) \tag{P.5}$$

The optimal measure that satisfies eq. (P.4) is the Gibbs measure

$$\mathrm{d}\mathscr{L}^* = \frac{\exp(-rJ)\mathrm{d}\mathscr{L}}{\mathbb{E}_{\mathscr{L}}\big[\exp(-rJ)\big]}. \tag{P.6}$$

P.4 Variational Optimization

$$\begin{aligned}
\mathbf{u}^* &= \underset{\mathbf{u}}{\operatorname{argmin}}\, S_c\big(\mathscr{L}^* || \tilde{\mathscr{L}}(\mathbf{u})\big) \\
&= \underset{\mathbf{u}}{\operatorname{argmin}}\, \mathbb{E}_{\mathscr{L}^*}\left[\ln \frac{\mathrm{d}\mathscr{L}^*}{\mathrm{d}\tilde{\mathscr{L}}(\mathbf{u})}\right] \\
&= \underset{\mathbf{u}}{\operatorname{argmin}} \int_\Omega \ln\left(\frac{\mathrm{d}\mathscr{L}^*}{\mathrm{d}\mathscr{L}}\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}(\mathbf{u})}\right)\mathrm{d}\mathscr{L}^* \\
&= \int_\Omega \ln\left(\frac{\mathrm{d}\mathscr{L}^*}{\mathrm{d}\mathscr{L}}\right)\mathrm{d}\mathscr{L}^* + \underset{\mathbf{u}}{\operatorname{argmin}} \int_\Omega \ln\left(\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}(\mathbf{u})}\right)\mathrm{d}\mathscr{L}^* \\
&= \underset{\mathbf{u}}{\operatorname{argmin}} \int_\Omega \ln\left(\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}(\mathbf{u})}\right)\mathrm{d}\mathscr{L}^* \\
&= \underset{\mathbf{u}}{\operatorname{argmin}} \int_\Omega \ln\left(\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}(\mathbf{u})}\right)\frac{\mathrm{d}\mathscr{L}^*}{\mathrm{d}\mathscr{L}}\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\tilde{\mathscr{L}}(\mathbf{u})}\mathrm{d}\tilde{\mathscr{L}}(\mathbf{u})
\end{aligned} \tag{P.7}$$

Plugging in the Gibbs measure eq. (P.6) and the Radon-Nikodym derivative eq. (P.3)

into eq. (P.7) yields

$$\begin{aligned}
\mathbf{u}^* = \underset{\mathbf{u}}{\operatorname{argmin}} \int_\Omega \Bigg(& -\int_0^T \mathbf{u}^\top \mathrm{Tr}_\rho \Big\{ i\big[\mathbf{H}_u, \rho_s\big] B(\rho_s)^{-1} \mathrm{d}W_s \Big\} - \frac{1}{2} \int_0^T \mathbf{u}^\top \mathrm{Tr}_\rho \Big\{ \big[\mathbf{H}_u, \rho_s\big] B(\rho_s)^{-2} \big[\mathbf{H}_u^\top, \rho_s\big] \Big\} \mathbf{u} \, \mathrm{d}s \Bigg) \\
& \frac{\exp(-rJ)}{\mathbb{E}_{\mathscr{L}}\big[\exp(-rJ)\big]} \exp\Bigg(-\int_0^T \mathbf{u}^\top \mathrm{Tr}_\rho \Big\{ i\big[\mathbf{H}_u, \rho_s\big] B(\rho_s)^{-1} \mathrm{d}W_s \Big\} \\
& -\frac{1}{2} \int_0^T \mathbf{u}^\top \mathrm{Tr}_\rho \Big\{ \big[\mathbf{H}_u, \rho_s\big] B(\rho_s)^{-2} \big[\mathbf{H}_u^\top, \rho_s\big] \Big\} \mathbf{u} \, \mathrm{d}s \Bigg) \mathrm{d}\tilde{\mathscr{L}}(\mathbf{u}) \\
= \underset{\mathbf{u}}{\operatorname{argmin}} \mathbb{E}_{\tilde{\mathscr{L}}} \Bigg[& \Bigg(-\int_0^T \mathbf{u}^\top \mathrm{Tr}_\rho \Big\{ i\big[\mathbf{H}_u, \rho_s\big] B(\rho_s)^{-1} \mathrm{d}W_s \Big\} \\
& -\frac{1}{2} \int_0^T \mathbf{u}^\top \mathrm{Tr}_\rho \Big\{ \big[\mathbf{H}_u, \rho_s\big] B(\rho_s)^{-2} \big[\mathbf{H}_u^\top, \rho_s\big] \Big\} \mathbf{u} \, \mathrm{d}s \Bigg) \frac{\exp(-r\tilde{J})}{\mathbb{E}_{\tilde{\mathscr{L}}}\big[\exp(-rJ)\frac{\mathrm{d}\tilde{\mathscr{L}}}{\mathrm{d}\mathscr{L}}\big]} \Bigg] \\
= \underset{\mathbf{u}}{\operatorname{argmin}} \mathbb{E}_{\tilde{\mathscr{L}}} \Bigg[& \Bigg(-\int_0^T \mathbf{u}^\top \mathrm{Tr}_\rho \Big\{ i\big[\mathbf{H}_u, \rho_s\big] B(\rho_s)^{-1} \mathrm{d}W_s \Big\} \\
& -\frac{1}{2} \int_0^T \mathbf{u}^\top \mathrm{Tr}_\rho \Big\{ \big[\mathbf{H}_u, \rho_s\big] B(\rho_s)^{-2} \big[\mathbf{H}_u^\top, \rho_s\big] \Big\} \mathbf{u} \, \mathrm{d}s \Bigg) \frac{\exp(-r\tilde{J})}{\mathbb{E}_{\tilde{\mathscr{L}}}\big[\exp(-r\tilde{J})\big]} \Bigg],
\end{aligned} \tag{P.8}$$

where

$$\tilde{J} = J + \int_0^T \mathbf{u}^\top \mathrm{Tr}_\rho \Big\{ i\big[\mathbf{H}_u, \rho_s\big] B(\rho_s)^{-1} \mathrm{d}W_s \Big\} + \frac{1}{2} \int_0^T \mathbf{u}^\top \mathrm{Tr}_\rho \Big\{ \big[\mathbf{H}_u, \rho_s\big] B(\rho_s)^{-2} \big[\mathbf{H}_u^\top, \rho_s\big] \Big\} \mathbf{u} \, \mathrm{d}s. \tag{P.9}$$

Next, the minimization is evaluated piece by piece, where we pass gradients into integrals with a properly defined Dominated Convergence Theorem

$$\begin{aligned}
\frac{\partial}{\partial \mathbf{u}} \Bigg(-\int_0^T \mathbf{u}^\top \mathrm{Tr}_\rho \Big\{ i\big[\mathbf{H}_u, \rho_s\big] B(\rho_s)^{-1} \mathrm{d}W_s \Big\} \Bigg) &= -\int_0^T \frac{\partial}{\partial \mathbf{u}} \Bigg(\mathbf{u}^\top \mathrm{Tr}_\rho \Big\{ i\big[\mathbf{H}_u, \rho_s\big] B(\rho_s)^{-1} \mathrm{d}W_s \Big\} \Bigg) \\
&= -\int_0^T \mathrm{Tr}_\rho \Big\{ i\big[\mathbf{H}_u, \rho_s\big] B(\rho_s)^{-1} \mathrm{d}W_s \Big\}
\end{aligned} \tag{P.10}$$

$$
\begin{aligned}
&\frac{\partial}{\partial \mathbf{u}}\left(-\frac{1}{2}\int_0^T \mathbf{u}^\top \mathrm{Tr}_\rho\Big\{ \big[\mathbf{H}_u,\rho_s\big] B(\rho_s)^{-2}\big[\mathbf{H}_u^\top,\rho_s\big]\Big\}\mathbf{u}\,\mathrm{d}s\right)\\
&= -\frac{1}{2}\int_0^T \frac{\partial}{\partial \mathbf{u}}\left(\mathbf{u}^\top \mathrm{Tr}_\rho\Big\{ \big[\mathbf{H}_u,\rho_s\big] B(\rho_s)^{-2}\big[\mathbf{H}_u^\top,\rho_s\big]\Big\}\mathbf{u}\right)\mathrm{d}s\\
&= -\frac{1}{2}\int_0^T \mathrm{Tr}_\rho\Big\{ \big[\mathbf{H}_u,\rho_s\big] B(\rho_s)^{-2}\big[\mathbf{H}_u^\top,\rho_s\big]\Big\}\mathbf{u}\,\mathrm{d}s - \frac{1}{2}\int_0^T \mathrm{Tr}_\rho\Big\{ \big[\mathbf{H}_u,\rho_s\big] B(\rho_s)^{-2}\big[\mathbf{H}_u^\top,\rho_s\big]\Big\}\mathbf{u}\,\mathrm{d}s\\
&= -\int_0^T \mathrm{Tr}_\rho\Big\{ \big[\mathbf{H}_u,\rho_s\big] B(\rho_s)^{-2}\big[\mathbf{H}_u^\top,\rho_s\big]\Big\}\mathbf{u}\,\mathrm{d}s
\end{aligned}
\tag{P.11}
$$

Putting the pieces together yields

$$
\begin{aligned}
0 &= \frac{\partial}{\partial \mathbf{u}}\Bigg(\mathbb{E}_{\tilde{\mathscr{Q}}}\Bigg[\Bigg(-\int_0^T \mathbf{u}^\top \mathrm{Tr}_\rho\Big\{ i\big[\mathbf{H}_u,\rho_s\big] B(\rho_s)^{-1}\mathrm{d}W_s\Big\}\\
&\qquad\qquad -\frac{1}{2}\int_0^T \mathbf{u}^\top \mathrm{Tr}_\rho\Big\{ \big[\mathbf{H}_u,\rho_s\big] B(\rho_s)^{-2}\big[\mathbf{H}_u^\top,\rho_s\big]\Big\}\mathbf{u}\,\mathrm{d}s\Bigg)\frac{\exp(-r\tilde{J})}{\mathbb{E}_{\tilde{\mathscr{Q}}}\big[\exp(-r\tilde{J})\big]}\Bigg]\Bigg)\\
&= \mathbb{E}_{\tilde{\mathscr{Q}}}\Bigg[\frac{\partial}{\partial \mathbf{u}}\Bigg(\Bigg(-\int_0^T \mathbf{u}^\top \mathrm{Tr}_\rho\Big\{ i\big[\mathbf{H}_u,\rho_s\big] B(\rho_s)^{-1}\mathrm{d}W_s\Big\}\\
&\qquad\qquad -\frac{1}{2}\int_0^T \mathbf{u}^\top \mathrm{Tr}_\rho\Big\{ \big[\mathbf{H}_u,\rho_s\big] B(\rho_s)^{-2}\big[\mathbf{H}_u^\top,\rho_s\big]\Big\}\mathbf{u}\,\mathrm{d}s\Bigg)\frac{\exp(-r\tilde{J})}{\mathbb{E}_{\tilde{\mathscr{Q}}}\big[\exp(-r\tilde{J})\big]}\Bigg)\Bigg]\\
&= \mathbb{E}_{\tilde{\mathscr{Q}}}\Bigg[\Bigg(-\int_0^T \mathrm{Tr}_\rho\Big\{ i\big[\mathbf{H}_u,\rho_s\big] B(\rho_s)^{-1}\mathrm{d}W_s\Big\}\\
&\qquad\qquad -\int_0^T \mathrm{Tr}_\rho\Big\{ \big[\mathbf{H}_u,\rho_s\big] B(\rho_s)^{-2}\big[\mathbf{H}_u^\top,\rho_s\big]\Big\}\mathbf{u}\,\mathrm{d}s\Bigg)\frac{\exp(-r\tilde{J})}{\mathbb{E}_{\tilde{\mathscr{Q}}}\big[\exp(-r\tilde{J})\big]}\Bigg]
\end{aligned}
\tag{P.12}
$$

Since we apply control in discrete time, we approximate both terms as follows

$$\int_0^T \mathrm{Tr}_\rho\Big\{\big[\mathbf{H}_u,\rho_s\big]B(\rho_s)^{-2}\big[\mathbf{H}_u^\top,\rho_s\big]\Big\}\mathbf{u}\,\mathrm{d}s\Big) \approx \sum_{l=1}^{L-1}\int_{t_l}^{t_{l+1}} \mathrm{Tr}_\rho\Big\{\big[\mathbf{H}_u,\rho_s\big]B(\rho_s)^{-2}\big[\mathbf{H}_u^\top,\rho_s\big]\Big\}\mathrm{d}s\;\mathbf{u}_l \tag{P.13}$$

$$\int_0^T \mathrm{Tr}_\rho\Big\{\big[\mathbf{H}_u,\rho_s\big]B(\rho_s)^{-1}\mathrm{d}W_s\Big\} \approx \sum_{l=1}^{L-1}\int_{t_l}^{t_{l+1}} \mathrm{Tr}_\rho\Big\{\big[\mathbf{H}_u,\rho_s\big]B(\rho_s)^{-1}\mathrm{d}W_s\Big\} \tag{P.14}$$

Solving for the optimal control yields the update

$$\mathbf{u}_l^* = -\,\mathbb{E}_{\tilde{\mathscr{Q}}}\left[\int_{t_l}^{t_{l+1}} \mathrm{Tr}_\rho\Big\{\big[\mathbf{H}_u,\rho_s\big]B(\rho_s)^{-2}\big[\mathbf{H}_u^\top,\rho_s\big]\Big\}\mathrm{d}s\frac{\exp(-r\tilde{J})}{\mathbb{E}_{\tilde{\mathscr{Q}}}\big[\exp(-r\tilde{J})\big]}\right]^{-1}\mathbb{E}_{\tilde{\mathscr{Q}}}\left[\int_{t_l}^{t_{l+1}} \mathrm{Tr}_\rho\Big\{i\big[\mathbf{H}_u,\rho_s\big]B(\rho_s)^{-1}\mathrm{d}W_s\Big\}\frac{\exp(-r\tilde{J})}{\mathbb{E}_{\tilde{\mathscr{Q}}}\big[\exp(-r\tilde{J})\big]}\right] \tag{P.15}$$

where the inverse term is a matrix in the dimensionality of $\mathbf{u}_l$, but a scalar in the dimensionality of ρ_t. Likewise the second expectation term is a vector in the dimensionality of $\mathbf{u}_l$, but a scalar in the dimensionality of ρ_t.

APPENDIX Q

CONNECTION BETWEEN THE KUSHNER-STRATONOVICH EQUATION AND THE BELAVKIN EQUATION

In this section, we briefly explore the connection between the Belavkin equation in eqs. (7.12) and (7.13). This connection can also be found in [190]. Consider the classical finite dimensional dynamics with stochastic evolution as

$$\mathrm{d}x(t) = F(x,t)\mathrm{d}t + B(x,t)\mathrm{d}W(t), \tag{Q.1}$$

where $x \in \mathbb{R}^d$ is the state, F is the nonlinear function for the drift, B is the covariance of the diffusion, and $W(t)$ is a Wiener process on a properly defined probability triple $(\Omega, \mathscr{F}, \mathbb{P})$ with filtration $\mathscr{F}_t$. Let the observation process be given by

$$\mathrm{d}y(t) = H(x,t)\mathrm{d}t + R(t)\mathrm{d}V(t), \tag{Q.2}$$

where H is a potentially partially observable drift term, R is a covariance of the diffusion, and $V(t)$ is another Wiener process. One can obtain the Fokker-Planck PDE using the Kramers-Moyal expansion

$$\frac{\partial p(x,t)}{\partial t} = \sum_{i=1}^{\infty} \frac{(-1)^n}{n!} \frac{\partial^n}{\partial x^n} \big[\alpha_n(x) p(x,t)\big], \tag{Q.3}$$

where $p(x,t)$ is the normalized distribution of the state,

$$\alpha_n(x) := \int_{-\infty}^{\infty} (x' - x)^n W(x'|x)\mathrm{d}x', \tag{Q.4}$$

and $W(x'|x)$ is the transition probability rate. Truncating the Kramers-Moyal expansion to second order yields the Fokker-Planck equation

$$\frac{\partial p(x,t)}{\partial t} = -\sum_{i=1}^{d} \frac{\partial}{\partial x_i}\big[F_i(x,u)p(x,t)\big] + \frac{1}{2}\sum_{i,j=1}^{d} \frac{\partial^2}{\partial x_i \partial x_j}\Big(\big[B(x,u)B^{\top}(x,u)\big]_{i,j}p(x,t)\Big). \quad \text{(Q.5)}$$

The Fokker-Planck PDE describes the unconditional evolution of the system (i.e. where we have not incorporated the observation process). The Fokker-Planck equation can be analogously written in operator notation with the Fokker-Planck operator $\mathscr{A}$ as

$$\mathrm{d}p_t = \mathscr{A}\, p_t \mathrm{d}t. \quad \text{(Q.6)}$$

Conditioning the evolution on the measurement or observation yields the Zakai equation as

$$\mathrm{d}p_t = \mathscr{A}\, p_t \mathrm{d}t + p_t H^{\top} \mathrm{d}y, \quad \text{(Q.7)}$$

Note that the distribution $p(x,t)$ in eq. (Q.7) is no longer normalized. Normalizing the Zakai equation yields the Kushner-Stratonovich equation

$$\mathrm{d}p_t = \mathscr{A}\, p_t \mathrm{d}t + p_t\Big[H + \langle H\rangle\Big]^{\top}\big(RR^{\top}\big)^{-1}\mathrm{d}Z_t, \quad \text{(Q.8)}$$

$$\text{Innovation:} \quad \mathrm{d}Z_t = \mathrm{d}y_t - \langle H(x,t)\rangle \mathrm{d}t \quad \text{(Q.9)}$$

where $Z(t)$ is an innovation process, which is itself a stochastic process driven by the stochastic measurement process $y(t)$.

Now, for ease of comparison we re-write the Belavkin equation for continuous QND

measurement of an open quantum system in eqs. (7.12) and (7.13),

$$\mathrm{d}\rho_c^t = \mathscr{L}_0\rho_c^t\mathrm{d}t + \mathscr{D}[V]\rho_c^t\mathrm{d}t + \Big(V\rho_c^t + \rho_c^t V^\dagger - \mathrm{Tr}\big[(V+V^\dagger)\rho_c^t\big]\rho_c^t\Big)\mathrm{d}W_t \tag{Q.10}$$

$$\text{Innovation:} \quad \mathrm{d}W_t = \mathrm{d}y_t - \mathrm{Tr}\big[(V+V^\dagger)\rho_c^t\big]\mathrm{d}t. \tag{Q.11}$$

Thus one can draw similarities between the Kushner-Stratonovich equation eq. (Q.8) and the Belavkin equation eq. (Q.10). Namely, the unconditional evolution of Kushner-Stratonovich equation and the Belavkin are given, respectively, by $\mathscr{A}p(x,t)\mathrm{d}t$ and $\big(\mathscr{L}_0\rho_c^t + \mathscr{D}[V]\rho_c^t\big)\mathrm{d}t$. The conditioning terms lead to stochasticity, and are given as the terms multiplying $\mathrm{d}Z$ and $\mathrm{d}W$, respectively. Finally the innovation process in eq. (Q.9) and eq. (Q.11) each describe the difference between what you measure, given by $\mathrm{d}y$ in both cases, and what you expect to measure, given by $\langle H(x,t)\rangle$ in the classical case and $\langle V+V^\dagger\rangle = \mathrm{Tr}\big[(V+V^\dagger)\rho_c^t\big]$ in the quantum case. Thus, one can think of the Belavkin equation as the quantum filtering analog of the Kushner-Stratonovich equation in classical filtering.

REFERENCES

[1] P. Chow, *Stochastic Partial Differential Equations*, ser. Advances in Applied Mathematics. Taylor & Francis, 2007, ISBN: 9781584884439.

[2] G. Da Prato and J. Zabczyk, *Stochastic Equations in Infinite Dimensions*, ser. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2014, ISBN: 9780521385299.

[3] R. Mikulevicius and B. L. Rozovskii, "Stochastic navier–stokes equations for turbulent flows," *SIAM Journal on Mathematical Analysis*, vol. 35, no. 5, pp. 1250–1310, 2004. eprint: https://doi.org/10.1137/S0036141002409167.

[4] G. Dumont, A. Payeur, and A. Longtin, "A stochastic-field description of finite-size spiking neural networks," *PLOS Computational Biology*, vol. 13, no. 8, pp. 1–34, Aug. 2017.

[5] E. Pardouxt, "Stochastic partial differential equations and filtering of diffusion processes," *Stochastics*, vol. 3, no. 1-4, pp. 127–167, 1980. eprint: https://doi.org/10.1080/17442507908833142.

[6] O. Bang, P. L. Christiansen, F. If, K. Ø. Rasmussen, and Y. B. Gaididei, "Temperature effects in a nonlinear model of monolayer scheibe aggregates," *Phys. Rev. E*, vol. 49, pp. 4627–4636, 5 May 1994.

[7] R. Cont, "Modeling term structure dynamics: An infinite dimensional approach," *International Journal of Theoretical and Applied Finance*, vol. 08, no. 03, pp. 357–380, 2005. eprint: https://www.worldscientific.com/doi/pdf/10.1142/S0219024905003049.

[8] J. Gough, V. P. Belavkin, and O. G. Smolyanov, "Hamilton-jacobi-bellman equations for quantum optimal feedback control," *Journal of Optics B: Quantum and Semiclassical Optics*, vol. 7, no. 10, S237, 2005.

[9] L. Bouten, S. Edwards, and V. P. Belavkin, "Bellman equations for optimal feedback control of qubit states," *Journal of Physics B: Atomic, Molecular and Optical Physics*, vol. 38, no. 3, p. 151, 2005.

[10] K. Elamvazhuthi, H. Kuiper, and S. Berman, "Pde-based optimization for stochastic mapping and coverage strategies using robotic ensembles," *Automatica*, vol. 95, pp. 356–367, 2018.

[11] E. Aidman, V. Ivancevic, and A. Jennings, "A coupled reaction-diffusion field model for perception-action cycle with applications to robot navigation," *International Journal of Intelligent Defence Support Systems*, vol. 1, no. 2, pp. 93–115, 2008.

[12] Y. Shapiro, K. Gabor, and A. Wolf, "Modeling a hyperflexible planar bending actuator as an inextensible euler–bernoulli beam for use in flexible robots," *Soft Robotics*, vol. 2, no. 2, pp. 71–79, 2015.

[13] J. C. Ryu, F. C. Park, and Y. Y. Kim, "Mobile robot path planning algorithm by equivalent conduction heat flow topology optimization," *Structural and Multidisciplinary Optimization*, vol. 45, no. 5, pp. 703–715, 2012.

[14] G. Ferrari-Trecate, A. Buffa, and M. Gati, "Analysis of coordination in multi-agent systems through partial difference equations," *IEEE Transactions on Automatic Control*, vol. 51, no. 6, pp. 1058–1063, 2006.

[15] G. Da Prato, A. Debussche, and R. Temam, "Stochastic burgers' equation," *Nonlinear Differential Equations and Applications NoDEA*, vol. 1, no. 4, pp. 389–402, 1994.

[16] G. Fabbri, F. Gozzi, and A. Swiech, *Stochastic Optimal Control in Infinite Dimensions - Dynamic Programming and HJB Equations*, ser. Probability Theory and Stochastic Modelling 82. Springer, Jan. 2017, OS.

[17] Y. Lou, G. Hu, and P. D. Christofides, "Model predictive control of nonlinear stochastic pdes: Application to a sputtering process," in *2009 American Control Conference*, IEEE, 2009, pp. 2476–2483.

[18] S. N. Gomes, S. Kalliadasis, D. T. Papageorgiou, G. A. Pavliotis, and M. Pradas, "Controlling roughening processes in the stochastic kuramoto–sivashinsky equation," *Physica D: Nonlinear Phenomena*, vol. 348, pp. 33–43, 2017.

[19] G. D. Prato and A. Debussche, "Control of the stochastic burgers model of turbulence," *SIAM Journal on Control and Optimization*, vol. 37, no. 4, pp. 1123–1149, 1999. eprint: http://dx.doi.org/10.1137/S0363012996311307.

[20] S. J. Moura and H. K. Fathy, "Optimal boundary control of reaction–diffusion partial differential equations via weak variations," *Journal of Dynamic Systems, Measurement, and Control*, vol. 135, no. 3, p. 034 501, 2013.

[21] G. D. Prato and A. Debussche, "Control of the stochastic burgers model of turbulence," *SIAM Journal on Control and Optimization*, vol. 37, no. 4, pp. 1123–1149, 1999. eprint: http://dx.doi.org/10.1137/S0363012996311307.

[22] J. Feng, "Large deviation for diffusions and hamilton-jacobi equation in hilbert spaces," *Ann. Probab.*, vol. 34, no. 1, pp. 321–385, Jan. 2006.

[23] K. Bieker, S. Peitz, S. L. Brunton, J. N. Kutz, and M. Dellnitz, "Deep model predictive control with online learning for complex physical systems," *arXiv preprint arXiv:1905.10094*, 2019.

[24] A. G. Nair, C.-A. Yeh, E. Kaiser, B. R. Noack, S. L. Brunton, and K. Taira, "Cluster-based feedback control of turbulent post-stall separated flows," *Journal of Fluid Mechanics*, vol. 875, pp. 345–375, 2019.

[25] A. T. Mohan and D. V. Gaitonde, "A deep learning based approach to reduced order modeling for turbulent flow control using lstm neural networks," *arXiv preprint arXiv:1804.09269*, 2018.

[26] J. Morton, A. Jameson, M. J. Kochenderfer, and F. Witherden, "Deep dynamical modeling and control of unsteady fluid flows," in *Advances in Neural Information Processing Systems*, 2018, pp. 9258–9268.

[27] J. Rabault, M. Kuchta, A. Jensen, U. Réglade, and N. Cerardi, "Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control," *Journal of Fluid Mechanics*, vol. 865, pp. 281–302, 2019.

[28] S. Satheeshbabu, N. K. Uppalapati, G. Chowdhary, and G. Krishnan, "Open loop position control of soft continuum arm using deep reinforcement learning," in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 5133–5139.

[29] R. F. Curtain and K. Glover, "Robust stabilization of infinite dimensional systems by finite dimensional controllers," *Systems & control letters*, vol. 7, no. 1, pp. 41–47, 1986.

[30] M. Balas, "Feedback control of flexible systems," *IEEE Transactions on Automatic Control*, vol. 23, no. 4, pp. 673–679, Aug. 1978.

[31] A. Spielberg, A. Zhao, Y. Hu, T. Du, W. Matusik, and D. Rus, "Learning-in-the-loop optimization: End-to-end control and co-design of soft robots through learned deep latent representations," *Advances in Neural Information Processing Systems*, vol. 32, pp. 8284–8294, 2019.

[32] T. George Thuruthel, Y. Ansari, E. Falotico, and C. Laschi, "Control strategies for soft robotic manipulators: A survey," *Soft robotics*, vol. 5, no. 2, pp. 149–163, 2018.

[33] U. Boscain, G. Charlot, J.-P. Gauthier, S. Guérin, and H.-R. Jauslin, "Optimal control in laser-induced population transfer for two-and three-level quantum systems," *Journal of Mathematical Physics*, vol. 43, no. 5, pp. 2107–2132, 2002.

[34] P. Kumar, S. A. Malinovskaya, and V. S. Malinovsky, "Optimal control of population and coherence in three-level λ systems," *Journal of Physics B: Atomic, Molecular and Optical Physics*, vol. 44, no. 15, p. 154 010, 2011.

[35] D. Dong, C. Chen, T.-J. Tarn, A. Pechen, and H. Rabitz, "Incoherent control of quantum systems with wavefunction-controllable subspaces via quantum reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 4, pp. 957–962, 2008.

[36] D. Yang and J. Zhong, "Optimal actuator location of the minimum norm controls for stochastic heat equations," *arXiv preprint arXiv:1710.06079*, 2017.

[37] K. Lim, "Method for optimal actuator and sensor placement for large flexible structures," *Journal of Guidance, Control, and Dynamics*, vol. 15, no. 1, pp. 49–57, 1992.

[38] T. Nestorović and M. Trajkov, "Optimal actuator and sensor placement based on balanced reduced models," *Mechanical Systems and Signal Processing*, vol. 36, no. 2, pp. 271–289, 2013.

[39] D. Kasinathan and K. Morris, "H infinity optimal actuator location," *IEEE Transactions on Automatic Control*, vol. 58, no. 10, pp. 2522–2535, 2013.

[40] K. K. Chen and C. W. Rowley, "H 2 optimal actuator and sensor placement in the linearised complex ginzburg–landau system," *Journal of Fluid Mechanics*, vol. 681, pp. 241–260, 2011.

[41] K. Manohar, J. N. Kutz, and S. L. Brunton, "Optimal sensor and actuator placement using balanced model reduction," *arXiv preprint arXiv:1812.01574*, 2018.

[42] R. Grigoriev, M. Cross, and H. Schuster, "Pinning control of spatiotemporal chaos," *Physical Review Letters*, vol. 79, no. 15, p. 2795, 1997.

[43] S. Sinha, U. Vaidya, and R. Rajaram, "Optimal placement of actuators and sensors for control of nonequilibrium dynamics," in *2013 European Control Conference (ECC)*, IEEE, 2013, pp. 1083–1088.

[44] U. Vaidya, R. Rajaram, and S. Dasgupta, "Actuator and sensor placement in linear advection pde with building system application," *Journal of Mathematical Analysis and Applications*, vol. 394, no. 1, pp. 213–224, 2012.

[45] S. Amstutz and H. Andrä, "A new algorithm for topology optimization using a level-set method," *Journal of computational physics*, vol. 216, no. 2, pp. 573–588, 2006.

[46] Y. Lou and P. D. Christofides, "Optimal actuator/sensor placement for nonlinear control of the kuramoto-sivashinsky equation," *IEEE Transactions on Control Systems Technology*, vol. 11, no. 5, pp. 737–745, 2003.

[47] M. S. Edalatzadeh, D. Kalise, K. A. Morris, and K. Sturm, "Optimal actuator design for vibration control based on lqr performance and shape calculus," *arXiv preprint arXiv:1903.07572*, 2019.

[48] L. Pontryagin, V. Boltyanskii, R. Gamkrelidze, and E. Mishchenko, *The mathematical theory of Optimal Processes*. New York: Pergamon Press, 1962.

[49] R. Bellman and R. Kalaba, *Selected Papers On mathematical trends in Control Theory*. Dover Publications, 1964.

[50] J. Yong and X. Zhou, *Stochastic Controls: Hamiltonian Systems and HJB Equations*, ser. Stochastic Modelling and Applied Probability. Springer New York, 1999, ISBN: 9780387987231.

[51] E. Pardoux and A. Rascanu, *Stochastic Differential Equations, Backward SDEs, Partial Differential Equations*. Jul. 2014, vol. 69.

[52] W. H. Fleming and H. M. Soner, *Controlled Markov processes and viscosity solutions*, 2nd, ser. Applications of mathematics. New York: Springer, 2006.

[53] G. Williams, A. Aldrich, and E. A. Theodorou, "Model predictive path integral control: From theory to parallel computation," *Journal of Guidance, Control, and Dynamics*, vol. 40:2, pp. 344–357, 2017.

[54] I. Exarchos and E. A. Theodorou, "Stochastic optimal control via forward and backward stochastic differential equations and importance sampling," *Automatica*, vol. 87, pp. 159–165, 2018.

[55] E. Theodorou and E. Todorov, "Relative entropy and free energy dualities: Connections to path integral and kl control," in *the Proceedings of IEEE Conference on Decision and Control*, Dec. 2012, pp. 1466–1473.

[56] M. J. Wainwright, M. I. Jordan, *et al.*, "Graphical models, exponential families, and variational inference," *Foundations and Trends® in Machine Learning*, vol. 1, no. 1–2, pp. 1–305, 2008.

[57] E. A. Theodorou, G. I. Boutselis, and K. Bakshi, "Linearly solvable stochastic optimal control for infinite-dimensional systems," in *2018 IEEE Conference on Decision and Control (CDC)*, IEEE, 2018, pp. 4110–4116.

[58] V. Volterra, "Theory of functionals and of integral and integro-differential equations," 1959.

[59] E. N. Evans, O. So, A. P. Kendall, G.-H. Liu, and E. A. Theodorou, "Spatio-temporal differential dynamic programming for control of fields," *arXiv preprint arXiv:2104.04044*, 2021.

[60] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Aggressive driving with model predictive path integral control," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 1433–1440.

[61] ——, "Information theoretic model predictive control: Theory and applications to autonomous driving," *IEEE Transactions on Robotics (Conditionally accepted - arXiv:1707.02342)*, 2018.

[62] G. I. Boutselis, E. N. Evans, M. A. Pereira, and E. A. Theodorou, "Leveraging stochasticity for open loop and model predictive control of spatio-temporal systems," *Entropy*, vol. 23, no. 8, p. 941, 2021.

[63] E. N. Evans, M. A. Pereira, G. I. Boutselis, and E. A. Theodorou, "Variational optimization based reinforcement learning for infinite dimensional stochastic systems," in *Conference on Robot Learning*, 2019.

[64] E. N. Evans, A. P. Kendall, and E. A. Theodorou, "Stochastic spatio-temporal optimization for control and co-design of systems in robotics and applied physics," *arXiv preprint arXiv:2102.09144*, 2021.

[65] D. J. Griffiths and D. F. Schroeter, *Introduction to quantum mechanics*. Cambridge University Press, 2018.

[66] A. Barchielli and M. Gregoratti, *Quantum trajectories and measurements in continuous time: the diffusive case*. Springer, 2009, vol. 782.

[67] K. R. Parthasarathy, *An introduction to quantum stochastic calculus*. Birkhäuser, 2012, vol. 85.

[68] Y. Sakawa, "A matrix green's formula and optimal control of linear distributed-parameter systems," *Journal of Optimization Theory and Applications*, vol. 10, no. 5, pp. 290–299, 1972.

[69] L. C. Evans, "Partial differential equations and monge-kantorovich mass transfer," *Current developments in mathematics*, vol. 1997, no. 1, pp. 65–126, 1997.

[70] G. Da Prato and J. Zabczyk, *Stochastic Equations in Infinite Dimensions*, ser. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2014, ISBN: 9780521385299.

[71] I. Lasiecka and R. Triggiani, *Control theory for partial differential equations: continuous and approximation theories*. Cambridge University Press Cambridge, 2000, vol. 1.

[72] F. Tröltzsch, *Optimal control of partial differential equations: theory, methods, and applications*. American Mathematical Soc., 2010, vol. 112.

[73] M. I. Sumin, "The first variation and pontryagin's maximum principle in optimal control for partial differential equations," *Computational Mathematics and Mathematical Physics*, vol. 49, no. 6, pp. 958–978, 2009.

[74] J. M. Yong, "Pontryagin maximum principle for semilinear second order elliptic partial differential equations and variational inequalities with state constraints," *Differential and Integral Equations*, vol. 5, no. 6, pp. 1307–1334, 1992.

[75] Y. Tassa, N. Mansard, and E. Todorov, "Control-limited differential dynamic programming," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 1168–1175.

[76] Y. Aoyama, G. Boutselis, A. Patel, and E. A. Theodorou, "Constrained differential dynamic programming revisited," *arXiv preprint arXiv:2005.00985*, 2020.

[77] Y. Tassa, T. Erez, and W. D. Smart, "Receding horizon differential dynamic programming.," in *NIPS*, 2007, pp. 1465–1472.

[78] Y. Pan and E. Theodorou, "Probabilistic differential dynamic programming," *Advances in Neural Information Processing Systems*, vol. 27, pp. 1907–1915, 2014.

[79] Y. Pan, G. I. Boutselis, and E. A. Theodorou, "Efficient reinforcement learning via probabilistic trajectory optimization," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 11, pp. 5459–5474, 2018.

[80] W. Sun, E. A. Theodorou, and P. Tsiotras, "Game theoretic continuous time differential dynamic programming," in *2015 American Control Conference (ACC)*, IEEE, 2015, pp. 5593–5598.

[81] G. I. Boutselis, Y. Pan, and E. A. Theodorou, "Numerical trajectory optimization for stochastic mechanical systems," *SIAM Journal on scientific computing*, vol. 41, no. 4, A2065–A2087, 2019.

[82] G. I. Boutselis, G. De La Torre, and E. A. Theodorou, "Stochastic optimal control using polynomial chaos variational integrators," in *2016 American Control Conference (ACC)*, IEEE, 2016, pp. 6586–6591.

[83] S. Tzafestas and J. Nightingale, "Differential dynamic-programming approach to optimal nonlinear distributed-parameter control systems," in *Proceedings of the Institution of Electrical Engineers*, IET, vol. 116, 1969, pp. 1079–1084.

[84] D. H. Jacobson and D. Q. Mayne, *Differential dynamic programming*. New York: American Elsevier Pub. Co., 1970.

[85] C. Shoemaker and L. Liao, "Proof of the quadratic convergence of differential dynamic programming," Cornell University Operations Research and Industrial Engineering, Tech. Rep., 1990.

[86] S. Yakowitz and B. Rutherford, "Computational aspects of discrete-time optimal control," *Applied Mathematics and Computation*, vol. 15, no. 1, pp. 29–45, 1984.

[87] W. Sun, E. A. Theodorou, and P. Tsiotras, "Continuous-time differential dynamic programming with terminal constraints," in *2014 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, IEEE, 2014, pp. 1–6.

[88] S. Tzafestas and J. Nightingale, "Optimal control of a class of linear stochastic distributed-parameter systems," in *Proceedings of the Institution of Electrical Engineers*, IET, vol. 115, 1968, pp. 1213–1220.

[89] A. Friedman, *Partial differential equations of parabolic type*. Courier Dover Publications, 2008.

[90] J. D. O. Pantoja, "Algorithms for constrained optimization problems," *Differential Dynamic Programming and Newton's Method. International Journal of Control*, vol. 47, pp. 1539–1553, 1983.

[91] D. Murray and S. Yakowitz, "Differential dynamic programming and newton's method for discrete optimal control problems," *Journal of Optimization Theory and Applications*, vol. 43, no. 3, pp. 395–414, 1984.

[92] G. I. Boutselis and E. Theodorou, "Differential dynamic programming on lie groups: Derivation, convergence analysis and numerical results," *arXiv preprint arXiv:1809.07883*, 2018.

[93] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2012, pp. 4906–4913.

[94] D. Butusov, T. Karimov, and V. Ostrovskii, "Semi-implicit ode solver for matrix riccati equation," in *2016 IEEE NW Russia Young Researchers in Electrical and Electronic Engineering Conference (EIConRusNW)*, IEEE, 2016, pp. 168–172.

[95] G. J. Lord, C. E. Powell, and T. Shardlow, *An Introduction to Computational Stochastic PDEs*, ser. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2014.

[96] J. W. Demmel, S. C. Eisenstat, J. R. Gilbert, X. S. Li, and J. W. Liu, "A supernodal approach to sparse partial pivoting," *SIAM Journal on Matrix Analysis and Applications*, vol. 20, no. 3, pp. 720–755, 1999.

[97] I. V. Oseledets, "Tensor-train decomposition," *SIAM Journal on Scientific Computing*, vol. 33, no. 5, pp. 2295–2317, 2011.

[98] E. Todorov, "Efficient computation of optimal actions," *Proceedings of the national academy of sciences*, vol. 106, no. 28, pp. 11 478–11 483, 2009.

[99] E. A. Theodorou, "Nonlinear stochastic control and information theoretic dualities: Connections, interdependencies and thermodynamic interpretations," *Entropy*, vol. 17, no. 5, p. 3352, 2015.

[100] H. J. Kappen, "Path integrals and symmetry breaking for optimal control theory," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 11, P11011, 2005.

[101] B. Maslowski, "Stability of semilinear equations with boundary and pointwise noise," *Annali della Scuola Normale Superiore di Pisa - Classe di Scienze*, vol. Ser. 4, 22, no. 1, pp. 55–93, 1995.

[102] A. Debussche, M. Fuhrman, and G. Tessitore, "Optimal control of a stochastic heat equation with boundary-noise and boundary-control," *ESAIM: Control, Optimisation and Calculus of Variations*, vol. 13, no. 1, pp. 178–205, 2007.

[103] E. N. Evans, A. P. Kendall, G. I. Boutselis, and E. A. Theodorou, "Spatio-temporal stochastic optimization: Theory and applications to optimal control and co-design," in *Proceedings of Robotics: Science and Systems*, 2020.

[104] H. J. Kappen and H. C. Ruiz, "Adaptive importance sampling for control and inference," *Journal of Statistical Physics*, vol. 162, no. 5, pp. 1244–1266, Mar. 2016.

[105] D.-T. Jeng, "Forced model equation for turbulence," *The Physics of Fluids*, vol. 12, no. 10, pp. 2006–2010, 1969.

[106] T. E. Duncan, B. Maslowski, and B. Pasik-Duncan, "Ergodic boundary/point control of stochastic semilinear systems," *SIAM journal on control and optimization*, vol. 36, no. 3, pp. 1020–1047, 1998.

[107] E. Theodorou, "Nonlinear stochastic control and information theoretic dualities: Connections, interdependencies and thermodynamic interpretations," *Entropy*, vol. 17, no. 5, pp. 3352–3375, 2015.

[108] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Aggressive driving with model predictive path integral control," *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1433–1440, 2016.

[109] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[110] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[111] K. Chellapilla, S. Puri, and P. Simard, "High performance convolutional neural networks for document processing," 2006.

[112] Martin Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Y. Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, *TensorFlow: Large-scale machine learning on heterogeneous systems*, Software available from tensorflow.org, 2015.

[113] E. Zhou and J. Hu, "Gradient-based adaptive stochastic search for non-differentiable optimization," *IEEE Transactions on Automatic Control*, vol. 59, no. 7, pp. 1818–1832, 2014.

[114] F.-F. Jin and B.-Z. Guo, "Lyapunov approach to output feedback stabilization for the euler–bernoulli beam equation with boundary input disturbance," *Automatica*, vol. 52, pp. 95–102, 2015.

[115] F. Renda, M. Giorelli, M. Calisti, M. Cianchetti, and C. Laschi, "Dynamic model of a multibending soft robot arm driven by cables," *IEEE Transactions on Robotics*, vol. 30, no. 5, pp. 1109–1122, 2014.

[116] R. J. Webster III and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1661–1683, 2010.

[117] W. S. Rone and P. Ben-Tzvi, "Continuum robot dynamics utilizing the principle of virtual power," *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 275–287, 2013.

[118] I. S. Godage, G. A. Medrano-Cerda, D. T. Branson, E. Guglielmino, and D. G. Caldwell, "Dynamics for variable length multisection continuum arms," *The International Journal of Robotics Research*, vol. 35, no. 6, pp. 695–722, 2016.

[119] V. Falkenhahn, T. Mahl, A. Hildebrandt, R. Neumann, and O. Sawodny, "Dynamic modeling of bellows-actuated continuum robots using the euler–lagrange formalism," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1483–1496, 2015.

[120] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, no. 7553, pp. 467–475, 2015.

[121] G. S. Chirikjian, "Hyper-redundant manipulator dynamics: A continuum approximation," *Advanced Robotics*, vol. 9, no. 3, pp. 217–243, 1994.

[122] M. W. Hannan and I. D. Walker, "Kinematics and the implementation of an elephant's trunk manipulator and other continuum style robots," *Journal of robotic systems*, vol. 20, no. 2, pp. 45–63, 2003.

[123] H. Mochiyama, "Hyper-flexible robotic manipulators," in *IEEE International Symposium on Micro-NanoMechatronics and Human Science, 2005*, IEEE, 2005, pp. 41–46.

[124] G. S. Chirikjian and J. W. Burdick, "The kinematics of hyper-redundant robot locomotion," *IEEE transactions on robotics and automation*, vol. 11, no. 6, pp. 781–793, 1995.

[125] J. Till, C. E. Bryson, S. Chung, A. Orekhov, and D. C. Rucker, "Efficient computation of multiple coupled cosserat rod models for real-time simulation and control of parallel continuum manipulators," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2015, pp. 5067–5074.

[126] J. Till, V. Aloi, and C. Rucker, "Real-time dynamics of soft and continuum robots based on cosserat rod models," *The International Journal of Robotics Research*, vol. 38, no. 6, pp. 723–746, 2019.

[127] D. Trivedi, A. Lotfi, and C. D. Rahn, "Geometrically exact models for soft robotic manipulators," *IEEE Transactions on Robotics*, vol. 24, no. 4, pp. 773–780, 2008.

[128] T. Zheng, D. T. Branson, R. Kang, M. Cianchetti, E. Guglielmino, M. Follador, G. A. Medrano-Cerda, I. S. Godage, and D. G. Caldwell, "Dynamic continuum arm model for use with underwater robotic manipulators inspired by octopus vulgaris," in *2012 IEEE International Conference on Robotics and Automation*, IEEE, 2012, pp. 5289–5294.

[129] Y. Yekutieli, R. Sagiv-Zohar, R. Aharonov, Y. Engel, B. Hochner, and T. Flash, "Dynamic model of the octopus arm. i. biomechanics of the octopus reaching movement," *Journal of neurophysiology*, vol. 94, no. 2, pp. 1443–1458, 2005.

[130] I. D. Walker, D. M. Dawson, T. Flash, F. W. Grasso, R. T. Hanlon, B. Hochner, W. M. Kier, C. C. Pagano, C. D. Rahn, and Q. M. Zhang, "Continuum robot arms inspired by cephalopods," in *Unmanned Ground Vehicle Technology VII*, International Society for Optics and Photonics, vol. 5804, 2005, pp. 303–314.

[131] O. Etzmuss, J. Gross, and W. Strasser, "Deriving a particle system from continuum mechanics for the animation of deformable objects," *IEEE Transactions on Visualization and Computer Graphics*, vol. 9, no. 4, pp. 538–550, 2003.

[132] G. I. Boutselis, M. A. Pereira, E. N. Evans, and E. A. Theodorou, "Variational optimization for distributed and boundary control of stochastic fields," *arXiv preprint arXiv:1904.02274*, 2019.

[133] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[134] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[135] S. Boyd, N. Parikh, and E. Chu, *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.

[136] S. Curci, D. C. Mocanu, and M. Pechenizkiyi, "Truly sparse neural networks at scale," *arXiv preprint arXiv:2102.01732*, 2021.

[137] B. Liu, M. Wang, H. Foroosh, M. Tappen, and M. Pensky, "Sparse convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 806–814.

[138] E. Elsen, M. Dukhan, T. Gale, and K. Simonyan, "Fast sparse convnets," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 14 629–14 638.

[139] H. Schaeffer, R. Caflisch, C. D. Hauck, and S. Osher, "Sparse dynamics for partial differential equations," *Proceedings of the National Academy of Sciences*, vol. 110, no. 17, pp. 6634–6639, 2013.

[140] U. Boscain, T. Chambrion, and G. Charlot, "Nonisotropic 3-level quantum systems: Complete solutions for minimum time and minimum energy," *arXiv preprint quant-ph/0409022*, 2004.

[141] U. Boscain and P. Mason, "Time minimal trajectories for a spin 1/ 2 particle in a magnetic field," *Journal of Mathematical Physics*, vol. 47, no. 6, p. 062 101, 2006.

[142] A. Carlini, A. Hosoya, T. Koike, and Y. Okudaira, "Time-optimal quantum evolution," *Physical review letters*, vol. 96, no. 6, p. 060 503, 2006.

[143] P. Salamon, K. H. Hoffmann, and A. Tsirlin, "Optimal control in a quantum cooling problem," *Applied Mathematics Letters*, vol. 25, no. 10, pp. 1263–1266, 2012.

[144] U. Boscain, F. Grönberg, R. Long, and H. Rabitz, "Minimal time trajectories for two-level quantum systems with two bounded controls," *Journal of Mathematical Physics*, vol. 55, no. 6, p. 062 106, 2014.

[145] R. Romano, "Geometric analysis of minimum-time trajectories for a two-level quantum system," *Physical Review A*, vol. 90, no. 6, p. 062 302, 2014.

[146] F. Albertini and D. D'Alessandro, "Time optimal simultaneous control of two level quantum systems," *Automatica*, vol. 74, pp. 55–62, 2016.

[147] O. V. Morzhin and A. N. Pechen, "Krotov method for optimal control of closed quantum systems," *Russian Mathematical Surveys*, vol. 74, no. 5, p. 851, 2019.

[148] T. Szakács, B. Amstrup, P. Gross, R. Kosloff, H. Rabitz, and A. Lörincz, "Locking a molecular bond: A case study of csi," *Physical Review A*, vol. 50, no. 3, p. 2540, 1994.

[149] I. R. Sola, J. Santamaria, and D. J. Tannor, "Optimal control of multiphoton excitation: A black box or a flexible toolkit?" *The Journal of Physical Chemistry A*, vol. 102, no. 23, pp. 4301–4309, 1998.

[150] A. Bartana, R. Kosloff, and D. J. Tannor, "Laser cooling of molecules by dynamically trapped states," *Chemical Physics*, vol. 267, no. 1-3, pp. 195–207, 2001.

[151] C. P. Koch, J. P. Palao, R. Kosloff, and F. Masnou-Seeuws, "Stabilization of ultracold molecules using optimal control theory," *Physical Review A*, vol. 70, no. 1, p. 013 402, 2004.

[152] J. P. Palao, R. Kosloff, and C. P. Koch, "Protecting coherence in optimal control theory: State-dependent constraint approach," *Physical Review A*, vol. 77, no. 6, p. 063 412, 2008.

[153] T. Caneva, M. Murphy, T. Calarco, R. Fazio, S. Montangero, V. Giovannetti, and G. E. Santoro, "Optimal control at the quantum speed limit," *Physical review letters*, vol. 103, no. 24, p. 240 501, 2009.

[154] R. Eitan, M. Mundt, and D. J. Tannor, "Optimal control with accelerated convergence: Combining the krotov and quasi-newton methods," *Physical Review A*, vol. 83, no. 5, p. 053 426, 2011.

[155] J. P. Palao, D. M. Reich, and C. P. Koch, "Steering the optimization pathway in the control landscape using constraints," *Physical Review A*, vol. 88, no. 5, p. 053 409, 2013.

[156] N. M., C. Koch, and D. Sugny, "Time optimization and state-dependent constraints in the quantum optimal control of molecular orientation," *Journal of Modern Optics*, vol. 61, no. 10, pp. 857–863, 2014.

[157] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, and S. J. Glaser, "Optimal control of coupled spin dynamics: Design of nmr pulse sequences by gradient ascent algorithms," *Journal of magnetic resonance*, vol. 172, no. 2, pp. 296–306, 2005.

[158] G. Jäger, D. M. Reich, M. H. Goerz, C. P. Koch, and U. Hohenester, "Optimal quantum control of bose-einstein condensates in magnetic microtraps: Comparison of gradient-ascent-pulse-engineering and krotov optimization schemes," *Physical Review A*, vol. 90, no. 3, p. 033 628, 2014.

[159] M. Y. Niu, S. Boixo, V. N. Smelyanskiy, and H. Neven, "Universal quantum control through deep reinforcement learning," *Quantum Information*, vol. 5, no. 1, pp. 1–8, 2019.

[160] K. Saeedi, S. Simmons, J. Z. Salvail, P. Dluhy, H. Riemann, N. V. Abrosimov, P. Becker, H.-J. Pohl, J. J. Morton, and M. L. Thewalt, "Room-temperature quantum bit storage exceeding 39 minutes using ionized donors in silicon-28," *Science*, vol. 342, no. 6160, pp. 830–833, 2013.

[161] R. Schmidt, A. Negretti, J. Ankerhold, T. Calarco, and J. T. Stockburger, "Optimal control of open quantum systems: Cooperative effects of driving and dissipation," *Physical review letters*, vol. 107, no. 13, p. 130 404, 2011.

[162] M. Abdelhafez, D. I. Schuster, and J. Koch, "Gradient-based optimal control of open quantum systems using quantum trajectories and automatic differentiation," *Physical Review A*, vol. 99, no. 5, p. 052 327, 2019.

[163] C. P. Koch, "Controlling open quantum systems: Tools, achievements, and limitations," *Journal of Physics: Condensed Matter*, vol. 28, no. 21, p. 213 001, 2016.

[164] V. V. Shende, S. S. Bullock, and I. L. Markov, "Synthesis of quantum-logic circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 6, pp. 1000–1010, 2006.

[165] J. Carolan, M. Mohseni, J. P. Olson, M. Prabhu, C. Chen, D. Bunandar, M. Y. Niu, N. C. Harris, F. N. Wong, M. Hochberg, *et al.*, "Variational quantum unsampling on a quantum photonic processor," *Nature Physics*, vol. 16, no. 3, pp. 322–327, 2020.

[166] J. M. Arrazola, T. R. Bromley, J. Izaac, C. R. Myers, K. Brádler, and N. Killoran, "Machine learning method for state preparation and gate synthesis on photonic quantum computers," *Quantum Science and Technology*, vol. 4, no. 2, p. 024 004, 2019.

[167] M. H. Goerz, "Optimizing robust quantum gates in open quantum systems," Ph.D. dissertation, 2015.

[168] J. P. Palao and R. Kosloff, "Quantum computing by an optimal control algorithm for unitary transformations," *Physical review letters*, vol. 89, no. 18, p. 188 301, 2002.

[169] ——, "Optimal control theory for unitary transformations," *Physical Review A*, vol. 68, no. 6, p. 062 308, 2003.

[170] C. Ahn, A. C. Doherty, and A. J. Landahl, "Continuous quantum error correction via quantum feedback control," *Physical Review A*, vol. 65, no. 4, p. 042 301, 2002.

[171] H.-P. Breuer, F. Petruccione, *et al.*, *The theory of open quantum systems*. Oxford University Press on Demand, 2002.

[172] G. Milburn and D. Walls, "Quantum nondemolition measurements via quadratic coupling," *Physical Review A*, vol. 28, no. 4, p. 2065, 1983.

[173] H. M. Wiseman and G. J. Milburn, "Quantum theory of optical feedback via homodyne detection," *Physical Review Letters*, vol. 70, no. 5, p. 548, 1993.

[174] D. F. Walls and G. J. Milburn, *Quantum optics*. Springer Science & Business Media, 2007.

[175] V. Belavkin, "Quantum diffusion, measurement and filtering i," *Theory of Probability & Its Applications*, vol. 38, no. 4, pp. 573–585, 1994.

[176] V. P. Belavkin, "Quantum stochastic calculus and quantum nonlinear filtering," *Journal of Multivariate analysis*, vol. 42, no. 2, pp. 171–201, 1992.

[177] A. Barchielli and V. Belavkin, "Measurements continuous in time and a posteriori states in quantum mechanics," *Journal of Physics A: Mathematical and General*, vol. 24, no. 7, p. 1495, 1991.

[178] V. Belavkin, "Measurement, filtering and control in quantum open dynamical systems," *Reports on Mathematical Physics*, vol. 43, no. 3, A405–A425, 1999.

[179] ——, "A new wave equation for a continuous nondemolition measurement," *Physics letters A*, vol. 140, no. 7-8, pp. 355–358, 1989.

[180] ——, "Towards the theory of control in observable quantum systems," *arXiv preprint quant-ph/0408003*, 2004.

[181] V. P. Belavkin and P. Staszewski, "C^*-algebraic generalization of relative entropy and entropy," in *Annales de l'IHP Physique theorique*, vol. 37, 1982, pp. 51–58.

[182] V. Belavkin, "A continuous counting observation and posterior quantum dynamics," *Journal of Physics A: Mathematical and General*, vol. 22, no. 23, p. L1109, 1989.

[183] S. C. Edwards and V. P. Belavkin, "Optimal quantum filtering and quantum feedback control," *arXiv preprint quant-ph/0506018*, 2005.

[184] V. P. Belavkin, "Nondemolition principle of quantum measurement theory," *Foundations of Physics*, vol. 24, no. 5, pp. 685–714, 1994.

[185] V. P. Belavkin, "Generalized uncertainty relations and efficient measurements in quantum systems," *Theoretical and Mathematical Physics*, vol. 26, no. 3, pp. 213–222, 1976.

[186] V. P. Belavkin and S. Edwards, "Quantum filtering and optimal control," in *Quantum Stochastics and Information: Statistics, Filtering and Control*, World Scientific, 2008, pp. 143–205.

[187] L. Bouten, S. Edwards, and V. Belavkin, "Bellman equations for optimal feedback control of qubit states," *Journal of Physics B: Atomic, Molecular and Optical Physics*, vol. 38, no. 3, p. 151, 2005.

[188] L. Bouten, M. Guta, and H. Maassen, "Stochastic schrödinger equations," *Journal of Physics A: Mathematical and General*, vol. 37, no. 9, p. 3189, 2004.

[189] V. P. Belavkin, "Theory of the control of observable quantum systems," *Autom. Remote Control*, pp. 178–188, 1983.

[190] A. C. Doherty, S. Habib, K. Jacobs, H. Mabuchi, and S. M. Tan, "Quantum feedback control and classical control theory," *Physical Review A*, vol. 62, no. 1, p. 012 105, 2000.

[191] D. A. Steck, K. Jacobs, H. Mabuchi, T. Bhattacharya, and S. Habib, "Quantum feedback control of atomic motion in an optical cavity," *Physical review letters*, vol. 92, no. 22, p. 223 004, 2004.

[192] J. K. Stockton, "Continuous quantum measurement of cold alkali-atom spins," Ph.D. dissertation, California Institute of Technology, 2007.

[193] N. Yamamoto and L. Bouten, "Quantum risk-sensitive estimation and robustness," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 92–107, 2009.

[194] C. Pellegrini, "Existence, uniqueness and approximation of a stochastic schrödinger equation: The diffusive case," *The Annals of Probability*, pp. 2332–2353, 2008.

[195] M. Mirrahimi and R. Van Handel, "Stabilizing feedback controls for quantum systems," *SIAM Journal on Control and Optimization*, vol. 46, no. 2, pp. 445–467, 2007.

[196] W. Verstraelen and M. Wouters, "Gaussian quantum trajectories for the variational simulation of open quantum-optical systems," *Applied Sciences*, vol. 8, no. 9, p. 1427, 2018.

[197] Z. Wang, O. So, K. Lee, and E. A. Theodorou, "Adaptive risk sensitive model predictive control with stochastic search," in *Learning for Dynamics and Control*, PMLR, 2021, pp. 510–522.

[198] T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," in *NeurIPS*, 2018.

[199] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots, and E. A. Theodorou, "Information theoretic mpc for model-based reinforcement learning," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, pp. 1714–1721.

[200] J. R. Johansson, P. D. Nation, and F. Nori, "Qutip: An open-source python framework for the dynamics of open quantum systems," *Computer Physics Communications*, vol. 183, no. 8, pp. 1760–1772, 2012.

[201] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, pp. 8026–8037, 2019.

[202] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade*, Springer, 2012, pp. 9–48.

[203] B. Tzen and M. Raginsky, "Neural stochastic differential equations: Deep latent gaussian models in the diffusion limit," *arXiv preprint arXiv:1905.09883*, 2019.

[204] X. Liu, S. Si, Q. Cao, S. Kumar, and C.-J. Hsieh, "Neural sde: Stabilizing neural ode networks with stochastic noise," *arXiv preprint arXiv:1906.02355*, 2019.